CSE527 Introduction to Computer Vision Homework 3

Due Tue Oct 20, 2015

Introduction

In this homework, you are required to use the MatConvNet Library. MatConvNet is a Matlab toolbox implementing Convolutional Neural Networks for computer vision applications. You can download and install it by following the instructions here. It also provides a very good tutorial for Convolutional Neural Networks.(click here). Read the tutorial first and then finish the homework.

1 CNN Basics (10 points)

1.1 Convolution

The most important feature for CNNs is to utilize weight sharing. Suppose the input image has $M * N * K$ dimensions, and there are K' different filters, each filter has $M_f * N_f * K$ dimensions, how can you calculate the output feature maps? Write down the formula. Then try it with MatConvNet. First read your own image, change it to gray image and use $im2single()$ to convert to single format. For the filter, use 2 3*3 sobel filters (one for x direction, one for y direction), use $single()$ to convert the filter to single format. Use $vl\_nnconv()$ to apply the filters to the image and get the feature map. Try different stride and pad settings. Submit the feature maps you compute.

1.2 Non-linearity

The convolution operator is a linear operator. In addition to that, there are several non-linear operators. The simplest non-linearity is obtained by following a linear filter by a non-linear gating function, applied identically to each component (i.e. point-wise) of a feature map. The simplest such function is the Rectified Linear Unit (ReLU). Apply $vl\_nnrelu()$ on the results you get from the last question. Submit the result images you get.

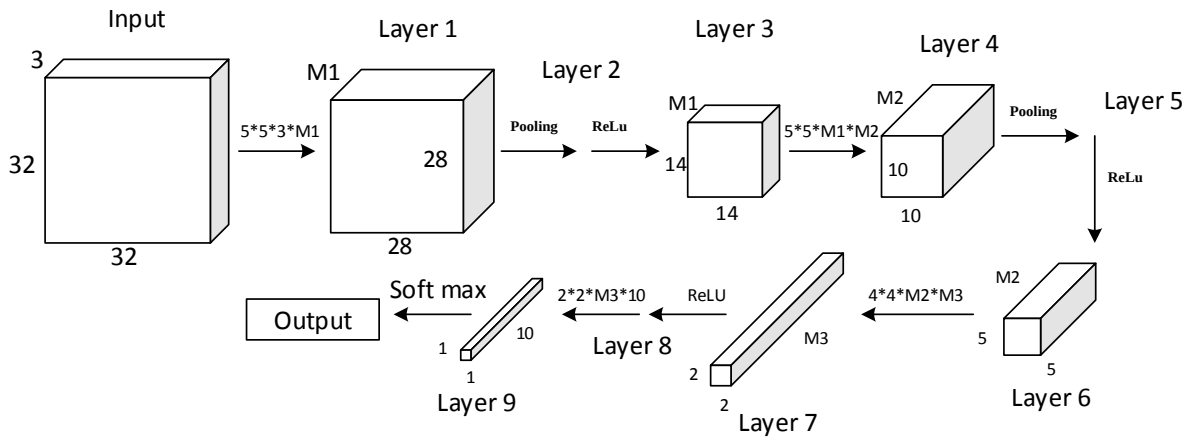
1.3 Pooling

What's the benefit of pooling? Apply $vl\_nnpool()$ on the results you get from last question. Try different stride and pad settings. Submit the result images you get.

2 Applying CNNs (40 points)

2.1 Using CNNs as a feature extractor and classifier (20 points)

In this question, you will apply CNNs to the image classification problem. The dataset is CIFAR 10 which consists of 60000 $32 * 32$ colour images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images. You can download the matlab version from here. Your model is the following:



You can refer to exercise4.m from the tutorial for the pipeline. You can directly use *cnn_train*() function. Tune the hyper-parameters eg. M1, M2, M3, batch size, epoch, learning rate, weightDecay, momentum until you think it performs well. Plot the error curve on both the training set and the test set. Test set visualization is for educational purposes. No decisions should be made based on test set performance. Visualize the filters from the first convolutional layer. Report the best accuracy you get on both training and testing sets.

2.2 Using CNN features and an SVM classifier (20 points)

After you train the network, use the network as a feature extractor. Feed each image to the network and use the feature maps before the last convolutional layer as the feature for the image (i.e. the feature maps from layer 8 in the above image). Extract such features for both training set and testing set. Train an SVM on training set features. Evaluate the accuracy on the training and testing datasets.

3 Transfer Learning (50 points)

AlexNet [1] is the winner of ILSVRC2012 Challenge. You should read the paper for details. In this problem, you need to apply the pre-trained AlexNet Model on the ImageNet Dataset to the Caltech 101 dataset. Caltech 101 has 101 classes of images (You can see the description and download from here). You should crop and resize the images to 224*224 because the input image to AlexNet should be 224*224. Use *cropImage*() function for cropping and resizing. For each class, you should use 30 images to train and 20 images to test. Make sure there is no overlap between training images and testing images. You can download AlexNet from here. Feed an image to AlexNet and use the feature maps before the last fully connected layer as features for the image (that is layer 19 in the pre-trained model). Use SVM as a classifier. Report your accuracy on training and testing.

Submission

Submission will be via Blackboard. You will submit the following files for each problem. Please submit a compressed file which includes all of them.

- Problem 1: Figures from step 1 and 3. Well documented code.
- Problem 2: Error curve during training. Figure of first convolutional layer filters. Well documented code.
- Problem 3: Well documented code. Write down your accuracy in report.

Plus, a report file containing your answer or description for each question (PDF format preferred). Please clarify which files are for which question (by writing in the report, organizing the folders, etc.) You may include images as figures in the report document instead of independent files.

References

[1] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Advances in neural information processing systems. 2012.