**请依次执行1,2,3,6这四条命令。**

1. **create\_lmdb.sh**

#!/usr/bin/env sh

# This script converts the mnist data into lmdb/leveldb format,

# depending on the value assigned to $BACKEND.

set -e

DATA=/home/hhj/caffe-master/examples/mnist

BUILD=/home/hhj/caffe-master/build/tools

rm -rf $DATA/mnist\_train\_lmdb

rm -rf $DATA/mnist\_test\_lmdb

$BUILD/convert\_imageset --shuffle \

--resize\_height=28 --resize\_width=28 \

$DATA/ \

$DATA/training.txt $DATA/mnist\_train\_lmdb

$BUILD/convert\_imageset --shuffle \

--resize\_height=28 --resize\_width=28 \

$DATA/ \

$DATA/testing.txt $DATA/mnist\_test\_lmdb

**2.compute\_mean.sh**

#!/usr/bin/env sh

# This script converts the mnist data into lmdb/leveldb format,

# depending on the value assigned to $BACKEND.

set -e

DATA=/home/hhj/caffe-master/examples/mnist

BUILD=/home/hhj/caffe-master/build/tools

rm -rf $DATA/mean.binaryproto

$BUILD/compute\_image\_mean $DATA/mnist\_train\_lmdb $DATA/mean.binaryproto $@

**3.train\_lenet.sh**

#!/usr/bin/env sh

set -e

BUILD=/home/hhj/caffe-master/build/tools

DATA=/home/hhj/caffe-master/examples/mnist

$BUILD/caffe train --solver=$DATA/lenet\_solver.prototxt $@

**4.lenet\_solver.prototxt**

# The train/test net protocol buffer definition

net: "/home/hhj/caffe-master/examples/mnist/lenet\_train\_test.prototxt"

# test\_iter specifies how many forward passes the test should carry out.

# In the case of MNIST, we have test batch size 100 and 100 test iterations,

# covering the full 10,000 testing images.

test\_iter: 100

# Carry out testing every 500 training iterations.

test\_interval: 500

# The base learning rate, momentum and the weight decay of the network.

base\_lr: 0.01

momentum: 0.9

weight\_decay: 0.0005

# The learning rate policy

lr\_policy: "inv"

gamma: 0.0001

power: 0.75

# Display every 100 iterations

display: 100

# The maximum number of iterations

max\_iter: 10000

# snapshot intermediate results

snapshot: 5000

snapshot\_prefix: "/home/hhj/caffe-master/examples/mnist/lenet"

# solver mode: CPU or GPU

solver\_mode: GPU

**5.lenet\_train\_test.prototxt**

1. name: "LeNet"
2. layer {
3. name: "mnist"
4. type: "Data"
5. top: "data"
6. top: "label"
7. include {
8. phase: TRAIN
9. }
10. transform\_param {
11. scale: 0.00390625
12. }
13. data\_param {
14. source: "/home/hhj/caffe-master/examples/mnist/mnist\_train\_lmdb"
15. batch\_size: 64
16. backend: LMDB
17. }
18. }
19. layer {
20. name: "mnist"
21. type: "Data"
22. top: "data"
23. top: "label"
24. include {
25. phase: TEST
26. }
27. transform\_param {
28. scale: 0.00390625
29. }
30. data\_param {
31. source: "/home/hhj/caffe-master/examples/mnist/mnist\_test\_lmdb"
32. batch\_size: 100
33. backend: LMDB
34. }
35. }
36. layer {
37. name: "conv1"
38. type: "Convolution"
39. bottom: "data"
40. top: "conv1"
41. param {
42. lr\_mult: 1
43. }
44. param {
45. lr\_mult: 2
46. }
47. convolution\_param {
48. num\_output: 20
49. kernel\_size: 5
50. stride: 1
51. weight\_filler {
52. type: "xavier"
53. }
54. bias\_filler {
55. type: "constant"
56. }
57. }
58. }
59. layer {
60. name: "pool1"
61. type: "Pooling"
62. bottom: "conv1"
63. top: "pool1"
64. pooling\_param {
65. pool: MAX
66. kernel\_size: 2
67. stride: 2
68. }
69. }
70. layer {
71. name: "conv2"
72. type: "Convolution"
73. bottom: "pool1"
74. top: "conv2"
75. param {
76. lr\_mult: 1
77. }
78. param {
79. lr\_mult: 2
80. }
81. convolution\_param {
82. num\_output: 50
83. kernel\_size: 5
84. stride: 1
85. weight\_filler {
86. type: "xavier"
87. }
88. bias\_filler {
89. type: "constant"
90. }
91. }
92. }
93. layer {
94. name: "pool2"
95. type: "Pooling"
96. bottom: "conv2"
97. top: "pool2"
98. pooling\_param {
99. pool: MAX
100. kernel\_size: 2
101. stride: 2
102. }
103. }
104. layer {
105. name: "ip1"
106. type: "InnerProduct"
107. bottom: "pool2"
108. top: "ip1"
109. param {
110. lr\_mult: 1
111. }
112. param {
113. lr\_mult: 2
114. }
115. inner\_product\_param {
116. num\_output: 500
117. weight\_filler {
118. type: "xavier"
119. }
120. bias\_filler {
121. type: "constant"
122. }
123. }
124. }
125. layer {
126. name: "relu1"
127. type: "ReLU"
128. bottom: "ip1"
129. top: "ip1"
130. }
131. layer {
132. name: "ip2"
133. type: "InnerProduct"
134. bottom: "ip1"
135. top: "ip2"
136. param {
137. lr\_mult: 1
138. }
139. param {
140. lr\_mult: 2
141. }
142. inner\_product\_param {
143. num\_output: 10
144. weight\_filler {
145. type: "xavier"
146. }
147. bias\_filler {
148. type: "constant"
149. }
150. }
151. }
152. layer {
153. name: "accuracy"
154. type: "Accuracy"
155. bottom: "ip2"
156. bottom: "label"
157. top: "accuracy"
158. include {
159. phase: TEST
160. }
161. }
162. layer {
163. name: "loss"
164. type: "SoftmaxWithLoss"
165. bottom: "ip2"
166. bottom: "label"
167. top: "loss"
168. include {
169. phase: TRAIN
170. }
171. }

**6.test\_lenet.sh**

#!/usr/bin/env sh

set -e

BUILD=/home/hhj/caffe-master/build/tools

DATA=/home/hhj/caffe-master/examples/mnist

$BUILD/caffe test -model $DATA/lenet\_train\_test.prototxt -weights $DATA/lenet\_iter\_10000.caffemodel -iterations 100 $@