

 chiaweil / dl4j-test

Branch: master ▾

Find file

Copy path

[dl4j-test](#) / [dl4j-test](#) / [src](#) / [main](#) / [java](#) / [org](#) / [deeplearning4j](#) / [convolution](#) / [ObjectClassification.java](#)

chiaweil added tests

ce7ac94 a day ago

1 contributor

159 lines (119 sloc) 5.5 KB

```
1 package org.deeplearning4j.convolution;
2
3 import org.datavec.api.io.filters.BalancedPathFilter;
4 import org.datavec.api.io.labels.ParentPathLabelGenerator;
5 import org.datavec.api.split.FileSplit;
6 import org.datavec.api.split.InputSplit;
7 import org.datavec.image.loader.NativeImageLoader;
8 import org.datavec.image.recordreader.ImageRecordReader;
9 import org.deeplearning4j.datasets.datavec.RecordReaderDataSetIterator;
10 import org.deeplearning4j.eval.Evaluation;
11 import org.deeplearning4j.nn.conf.MultiLayerConfiguration;
12 import org.deeplearning4j.nn.multilayer.MultiLayerNetwork;
13 import org.deeplearning4j.optimize.listeners.ScoreIterationListener;
14 import org.deeplearning4j.util.ModelSerializer;
15 import org.nd4j.linalg.dataset.api.iterator.DataSetIterator;
16 import org.nd4j.linalg.dataset.api.preprocessor.DataNormalization;
17 import org.nd4j.linalg.dataset.api.preprocessor.ImagePreProcessingScaler;
18 import org.nd4j.linalg.io.ClassPathResource;
19 import org.slf4j.Logger;
20 import org.slf4j.LoggerFactory;
21
22 import java.io.File;
23 import java.util.Random;
24
25 /**
26  * Using Convolutional Neural Network for classification of three classes [dog, donut, earth]
27  *
28  * Search for the part with *Enter your code here* and replace with model configuration
29  *
30  * [NOTE: Do not change other parts other than function getConfig(...)]
31  */
32 public class ObjectClassification
33 {
34     private static final Logger log = LoggerFactory.getLogger(ObjectClassification.class);
35
36     protected static int height = 100;           //Image height
37     protected static int width = 100;            //Image width
38     protected static int channels = 3;            //Image depth
39     protected static int classes = 3;            //Number of classes
40     protected static int batchSize = 25;
41
42     protected static long seed = 42;              //Seed number for reproduction
43     protected static Random rng = new Random(seed);
44
45     protected static double trainDataRatio = 0.8; //Segregate data into training and testing dataset
46
47     protected static int epochs = 20;            //Number of epochs
48
49     public void run() throws Exception
50     {
51         //Load model if exist. Test on single image
52         String rootPath = new ClassPathResource("/classification").getFile().toString();
53         File saveAs = new File(rootPath + "/objectModel.zip");
54     }
```

```
55
56 //Set image root directory
57 File rootDir = new File(rootPath + "/threeobjects");
58 FileSplit fileSplit = new FileSplit(rootDir, NativeImageLoader.ALLOWED_FORMATS, rng);
59
60 if(rootDir.exists() == false)
61 {
62
63     System.out.println("File not exist. Abort");
64     return;
65 }
66
67 //Get images directory' name as label
68 ParentPathLabelGenerator labelMaker = new ParentPathLabelGenerator();
69
70 //Get number of labels by number of directory in the rootDir. The rootDir must not contain other contents.
71 System.out.println(rootDir.toString());
72 int numLabels = rootDir.listFiles(File::isDirectory).length;
73
74 //Split into training and testing file split, images of different labels shuffled here
75 BalancedPathFilter pathFilter = new BalancedPathFilter(new Random(seed), NativeImageLoader.ALLOWED_FORMATS, labelMaker);
76
77 InputSplit[] inputSplit = fileSplit.sample(pathFilter, trainDataRatio, 1 - trainDataRatio);
78 InputSplit trainSplit = inputSplit[0];
79 InputSplit testSplit = inputSplit[1];
80
81 //Set image record reader for training and testing data
82 ImageRecordReader rrTrain = new ImageRecordReader(height, width, channels, labelMaker);
83 rrTrain.initialize(trainSplit);
84
85 ImageRecordReader rrTest = new ImageRecordReader(height, width, channels, labelMaker);
86 rrTest.initialize(testSplit);
87
88 //Set data iterator
89 DataSetIterator iterTrain = new RecordReaderDataSetIterator(rrTrain, batchSize, 1, numLabels);
90 DataSetIterator iterTest = new RecordReaderDataSetIterator(rrTest, batchSize, 1, numLabels);
91
92
93 //Data normalization
94 DataNormalization scaler = new ImagePreProcessingScaler(0, 1);
95 scaler.fit(iterTrain);
96 iterTrain.setPreProcessor(scaler);
97 iterTest.setPreProcessor(scaler);
98
99
100 MultilayerConfiguration config = getConfig(channels, classes);
101
102
103 if(config == null)
104 {
105     System.out.println("Configuration not set right. Abort");
106     return;
107 }
108
109 //Build model
110 log.info("Build model...");
111
112 MultilayerNetwork network = new MultilayerNetwork(config);
113 network.init();
114 network.setListeners(new ScoreIterationListener(10));
115
116 //Start training
117 log.info("Train model...");
118
119 for (int i = 0; i < epochs; ++i)
120 {
121     network.fit(iterTrain);
```

```
122     }
123
124     //Evaluate model
125     Evaluation eval = network.evaluate(iterTest);
126     log.info(eval.stats());
127
128     //Save model
129     ModelSerializer.writeModel(network, saveAs, false);
130
131     log.info("Program end.");
132 }
133
134
135 /**
136  * Build network configuration
137  *
138  * @param numInputs input layer nodes
139  * @param numOutputs output layer nodes
140  * @return MultilayerConfiguration with network configuration
141  */
142 public static MultilayerConfiguration getConfig(int numInputs, int numOutputs) {
143
144     /**
145      * Enter your code here
146      */
147
148     return null; //change to return MultilayerConfiguration instance
149 }
150
151 public static void main(String[] args) throws Exception
152 {
153     ObjectClassification classifier = new ObjectClassification();
154
155     classifier.run();
156 }
157
158 }
```