University of Wollongong

School of Computing and Information Technology

CSCI446/946 **Big Data Analytics** **Spring 2020**

Assignment 3 **(Due: 24:00, 9th December 2020, Beijing Time)** **20 marks**

Aim

This assignment is intended to obtain experience in conducting image analytics (in R or any other languages preferred by students). After having completed this assignment you should know how to perform image classification by training a deep convolutional neural network or using a pre-trained deep CNN network.

Group work: You are to work as part of group on this assignment. Each group is to work independently from other groups on this assignment. You can form groups of your own accord. Each student can only join in one group. Each group should contain **no more than 4 members and no less than 2 members**. All group members are expected to contribute to this assignment. All your answers to this assignment must be accompanied with a justification and/or explanation. One submission per group only.

Penalties: If a group member fails to make a minimum in contributions, then the member will be awarded zero marks. Plagiarism of any part in this assignment will result in zero marks being awarded to the whole group.

Preliminaries

Read through the lecture notes and recommended readings on image analysis. Complete the tutorial and study all example programs therein so that you fully understand these techniques and know how to perform them with R (or any other languages you prefer). Important references include:

1. https://mxnet.incubator.apache.org/api/r
2. https://www.r-bloggers.com/image-recognition-tutorial-in-r-using-deep-convolutional-neural-networks-mxnet-package/
3. https://tensorflow.rstudio.com/
4. https://tensorflow.rstudio.com/tutorials/advanced/images/cnn/

You may also use other standard packages (i.e. pytorch, torchvision) for this assignment though mxnet will be required to work on task 3.

Task 1 – Understanding CNNs for image analysis (4 marks)

Deep convolutional neural networks (CNNs) achieve the state-of-the-art performance in image recognition. It is important to understand the basic working principles of deep CNNs in order to appropriately utilize them to resolve image analysis tasks. Study the lecture notes, recommended readings, and other learning resources in the Internet to **answer the following questions**.

1. Describe deep convolutional neural networks and discuss why they suit image analysis tasks; **(1 marks)**
2. Describe the functions of "convolution layer" and "pooling layer" in deep CNNs; **(1 marks)**
3. Explain the following concepts in training deep CNNs: a) activation function; b) epoch number; c) batch size; d) learning rate; and e) momentum. **(2 marks)**

Task 2 – Handwritten Digits Classification with CNNs (10 marks)

MNIST is a benchmark dataset for handwritten digits classification. Each sample in this dataset is a small image of the size 28 by 28. Each image belongs to one of the 10 categories corresponding to digits "0" to "9". Handwritten Digits Classification is to design image recognition algorithms that can best classify each image into the right category. This dataset has been pre-partitioned into training and test sets. Tutorials on how to deploy CNNs to this dataset can be obtained from the webpages:
MXNET: https://mxnet.incubator.apache.org/api/r/docs/tutorials/mnist_competition.
Tensorflow: https://tensorflow.rstudio.com/guide/tfestimators/examples/mnist/

Carefully read these webpages and obtain the training and test data sets. **Complete the following tasks**:

1. Reshape if necessary each image into a long vector of 784 (i.e., 28 x 28) dimensions. Train a logistic regression classifier (LRC) and a multi-class linear support vector machine (SVM) with the training data set and test it on the testing data set;

2. Use either the mxnet library or the tensorflow library to design and train a CNN by using the training data in such a way as to maximise its generalization capability. Refer to this trained model as CNNmodelA. Test your CNN on the testing data set;
3. Techniques such as dropout and image distortions have been suggested to encourage generalization capability of deep neural networks (read: https://machinelearningmastery.com/dropout-for-regularizing-deep-neural-networks/ , and http://people.idsia.ch/~ciresan/data/NNtricks.pdf). Suitably deploy dropout to create a CNNmodelB, and use image distortions as described in the NNtricks.pdf paper[1] to train another CNN to create CNNmodelC.
4. Evaluate and compare the classification performance of the three CNN models.
5. Evaluate and compare the classification performance of the CNNs with the results from the LRC and SVM.

In your report, you need to

1. Describe this MNIST data set and its training and test subsets.
2. Describe how you reshape each image into a long vector and how you train the LRC or SVM.
3. Describe how you designed your CNN, justify your approach, and describe how you trained it. Explain which other settings of the network parameters and/or training parameters that you tried, and describe the changes on classification accuracy and training time.
4. Report the best classification accuracy and the corresponding confusion matrices obtained by the classification methods (LRC, SVM, and the CNNs). Evaluate and compare the classification performances. Analyse and explain the results.
5. Attach your code to the ZIP file.

Task 3 – Image Classification with a Pre-trained CNN Model (6 marks)

Carefully read the following webpage
https://mxnet.apache.org/api/r/docs/tutorials/classify_real_image_with_pretrained_model and use the provided code to classify some object images. You are free to use any object images that are similar to those in the ImageNet dataset (http://www.image-net.org/challenges/LSVRC/).

After that, download (a reduced version of) the Caltech256 training and test sets from:
https://drive.google.com/drive/folders/0Bwnyd83DcfEdX0ppY0x4S293X00
Unzip the file and you will see 257 folders. Each folder corresponds to one class. Use the pre-trained "Inception-BatchNorm" network provided in the first webpage to extract feature representation of each image in the Caltech256 training and test sets. To do this you may find the discussion on following webpage helpful:
https://github.com/apache/incubator-mxnet/issues/2535
When feature representations are extracted for all images, train a logistic regression classifier (LRC) or a multi-class linear support vector machine (SVM) with the training data set and test it on the testing data set.

In your report, you need to

1. Describe how you use the pre-trained Inception-BatchNorm network to classify some object images.
2. Describe the provided Caltech256 training and test sets.
3. Describe how you use the pre-trained Inception-BatchNorm network to extract feature representation for each image in the Caltech256 dataset.
4. Report the best classification accuracy obtained by the LRC or SVM classifier. List the top 10 pairs of classes that are confused most in the confusion matrix.
5. Attach your code to the ZIP file.

Submit:

Important:

1. **The report must be in PDF format. The title page must list the full name and student ID of all members in the group. Specify the individual effort (in percentage) of your team members in the report.**
2. **The report must answer the questions in the order as defined in this assignment.**
3. **The report must have a clear heading for the part for each task.**
4. **The report shall contain sufficient and detailed description, explanation, justification and discussion. Marks will be deducted for an incomplete report.**
5. **Sufficient annotation shall be provided in your code to make it easy to understand.**

[1] You should distort the images by using random rotation, shear, scale, and elastic distortion as described in the paper. You do not need to implement the CUDA code or use any of the other "tricks" described in the paper.

Make sure your report and code are correctly formatted and titled. Marks will be deducted for untidy or incorrectly formatted work. Submit your report and the source code files in a zipped file named A3.zip via the submit link provided for Assignment 3 on the subjects' Moodle site. **Only one submission per group.**

Note: Plagiarism of any part of your code or report will attract zero marks. It is the responsibility of the group to ensure that your submission does not contain plagiarized material. Each group member may be requested to demonstrate and explain the code or report. Any group member who fails to demonstrate or explain the code or report would receive a penalty in marks. Marks will be awarded for correct design, reasoning, completeness, implementation, and style. Marks will be deducted for late submissions. The deduction will be 25% for each day (or parts thereof) late. Submissions more than three days late will not be assessed.

--- END ---