CS671 - Deep Learning and its Applications

Course Instructor : Aditya Nigam (Session : Feb-May 2017)

Assignment-03

Submission Date : April 25, 2017

Note:

1. Answer all questions. Maximum score is 200 points.
2. Make sure you clearly identify the question number and your final answer, and solve all parts of a question together.
3. You are free to make any reasonable assumption that you may need to logically answer a question.
4. Code has to be well commented and as general as possible.
5. You are expected to submit a make file to compile your code and a README file containing how to run your codes.

---

1. **The goal of this assignment is to learn the basic principles of designing deep convolutional neural networks for image classification**

   (a) Here you have been given biological cell images (HEp-2 cells) of seven different classes as shown in the Fig. 2 and the goal is to design a convolutional neural network that learns to predict from which class a particular image belongs too.

   (b) Data for seven classes have been given separately.The directory structure of the data is given as shown in the Fig. 1. For all classes, there are positive images and intermediate images. Positive images are good in contrast rather than intermediate images. You are advised to first understand the dataset carefully and then start implementation. In total 14 folders are given.Out of these 14 folders, 7 folders contains positive images and remaining 7 contains intermediate images.For example positive images of class1 are present in class1_pos folder and intermediate images of class1 are present in class1_sin folder, same is done for all the classes. Data specification for each class is given as shown in the Fig. 3(a). You have to take both positive and intermediate types images for each class. If your model is working well, then try the same model only on intermediate images so that the efficiency of the model can be judged on low contrast images also. Seven different classes are as follows:-

      i. Homogeneous(Class1)
      ii. Speckled(Class2)
      iii. Nucleolar(Class3)
      iv. Centromere(Class4)
      v. Golgi (Class5)
      vi. Nuclear Membrane(Class6)
      vii. Mitotic Spindle(Class7)

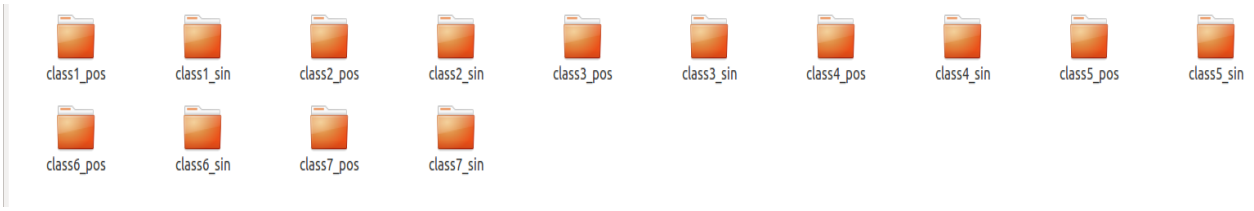
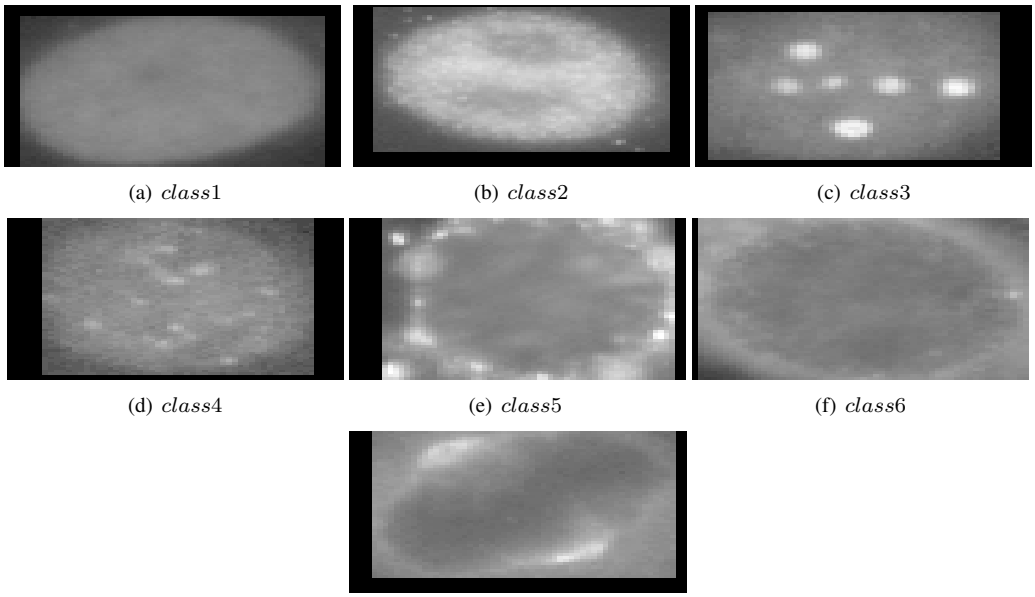


Figure 1: Directory Structure



(a) $class1$ (b) $class2$ (c) $class3$

(d) $class4$ (e) $class5$ (f) $class6$

(g) $class7$

Figure 2: Different Data Classes

(c) Out of the given data you can take 90 % images as the training data and the remaining 10 % images as the validation data. Note that we have a huge amount of training data for class1 to class6 but very less amount of training data for class7. This is due to the rare occurrence of this cell type. So your model should also do well for this specific case.

(d) Size of each image is different in different folders.You have to resize it accordingly before feeding the data to your CNN model.

(e) Your trained model will be checked on a test data(This data will be unseen to you).

(f) When the model is tested on the test data the output should contain the number of true positive, false positive, true negative and false negative samples of each class present in the given dataset(For example number of samples of Class1 correctly classified as Class1, number of samples of Class1 misclassified as Class2, misclassified as Class3,misclassified as Class4,misclassified as Class5,misclassified as Class6,misclassified as Class7.This has to be done for all the 7 classes).Output format for class1 is given as shown in Fig. 3(b).Same output format is followed for all the other six classes.

(g) You are free to chose any CNN model like VGGNet,GoogleNet,ResNet,AlexNet. Apart from that you can design your own CNN model too.

(h) The most important layers that you should include in your model are as follows:-

i. Convolutional Layer

ii. Pooling Layer (averagepool or mapool)as shown in Fig. 4(a)

iii. RELU Layer as shown in Fig. 4(b)

(i) You are free to chose hyperparameters like polling window size, stride window,batch size and number of iterations.

(j) You are free to chose python libraries for implementation.

(k) You have to make a report file in latex that includes classification accuracy for every class, precision for every class and F-measure for every class and overall accuracy.

i. **True Positive(TP)**: These are the cases in which ground truth and predicted value matches(say a sample belonging to classX is predicted as classX)

ii. **False Positive(FP)**:These are the number of samples that were classified say as belonging to the classX when they are actually members of some other class.

iii. **False Negative(FN)**:These are the number of samples that actually belongs to classX but are classified as not belonging to classX.

iv. **True Negative(TN)**:These are the cases in which ground truth and predicted value matches(say a sample not belonging to classX is predicted as not belongs to classX)

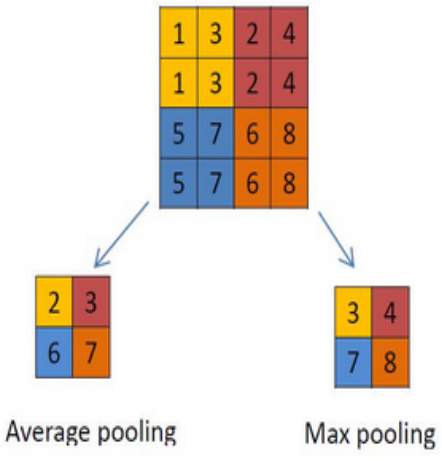
Class Name	Number of Samples(Images)
class1_pos	5012
class1_sin	4060
class2_sin	5012
class2_pos	5012
class3_sin	5012
class3_pos	5012
class4_sin	5012
class4_pos	5012
class5_sin	1008
class5_pos	720
class6_sin	2400
class6_pos	1200
class7_sin	70
class7_pos	40

(a) $DataSpecifications$

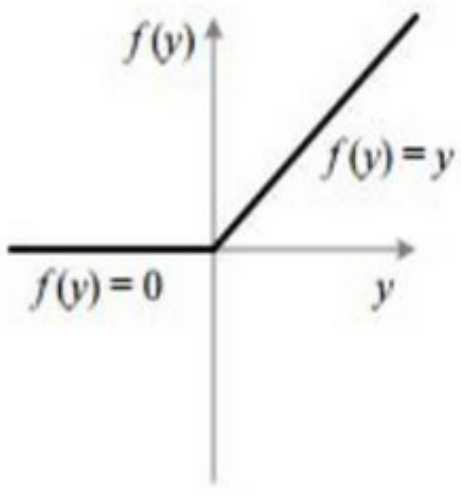
Evaluation Results
For Class1
No of samples Correctly Classified as Class1
No of samples of Class1 misclassified as Class2
No of samples of Class1 misclassified as Class3
No of samples of Class1 misclassified as Class4
No of samples of Class1 misclassified as Class5
No of samples of Class1 misclassified as Class6
No of samples of Class1 misclassified as Class7

(b) $OutputFormat$

Figure 3: (a)Data Specifications (b)Output Format



(a) $Pooling$



(b) $RELU$

Figure 4: (a)Pooling operation (b)RELU operation

v. **Classification Accuracy:** It tell us how much the classifier is able to classify the classes accurately.Classification accuracy is given as:-

$$Classification Accuracy = \left(\frac{TP + TN}{Total \ \ number \ \ of \ \ samples} \right) \tag{1}$$

vi. **Precision:**Given all the predicted labels (for a given class X), how many instances were correctly predicted.It is given as:-

$$Precision = \left(\frac{TP}{Total No. \ of \ \ samples \ \ predicted \ \ under \ \ that \ \ class} \right) \tag{2}$$

vii. **Recall or Sensitivity :** For all instances that should have a label X, how many of them were correctly classified.It is given as:-

$$Precision = \left(\frac{TP}{Total No. \ of \ \ samples \ \ actually \ \ belonging \ \ to \ \ that \ \ class} \right) \tag{3}$$

(l) Report should also include Confusion matrix based on the performance for validation data as shown in Fig. 5

		Predicted Class						
		Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7
Actual Class	Class 1							
	Class 2							
	Class 3							
	Class 4							
	Class 5							
	Class 6							
	Class 7							

Figure 5: Confusion Matrix

(m) It should also contain training accuray at each iteration.

(n) At the end plot the images that are misclassified in each class and try to find the reason behind this misclassification.