CELESTINI PROJECT INDIA 2018
TAKE-HOME EXAM
March 19, 2018

Authors:
Mayank Singh Chauhan - mayanksingh2298@gmail.com
Arshdeep Singh - arshdeepiitd@gmail.com

Link to our Github Repository : https://github.com/4rshdeep/celestini

1. Multiple-choice questions (10 points)

Select one or more correct solutions. Please write your answer next to **Solution:**

A.) What types of learning, if any, best describe the following three scenarios:
(i) A coin classification system is created for a vending machine. In order to do this, the developers obtain exact coin specifications from the U.S. Mint and derive a statistical model of the size, weight, and denomination, which the vending machine then uses to classify its coins.
(ii) Instead of calling the U.S. Mint to obtain coin information, an algorithm is presented with a large set of labeled coins. The algorithm uses this data to infer decision boundaries which the vending machine then uses to classify its coins.
(iii) A computer develops a strategy for playing Tic-Tac-Toe by playing repeatedly and adjusting its strategy by penalizing moves that eventually lead to losing.

[a] (i) Supervised Learning, (ii) Unsupervised Learning, (iii) Reinforcement Learning
[b] (i) Supervised Learning, (ii) Not learning, (iii) Unsupervised Learning
[c] (i) Not learning, (ii) Reinforcement Learning, (iii) Supervised Learning
[d] (i) Not learning, (ii) Supervised Learning, (iii) Reinforcement Learning
[e] (i) Supervised Learning, (ii) Reinforcement Learning, (iii) Unsupervised Learning
Solution: (d)

B.) For an imbalanced dataset, which of the following metric/tool is not that useful?
[a] F1 measure
[b] Accuracy
[c] Confusion Matrix
[d] Precision
Solution: (b)

C.) Consider the following implementation of a function mysteryFunction (pseudocode), where x is a positive integer:

```
mysteryFunction (x)
    xs = str(x)
    if len(xs) == 1
        return int(xs)
    n = int(xs[0]) + int(xs[1])
    if len(xs) == 2
        return n
    else
        return n + mysteryFunction(xs[2:])
```

What does mysteryFunction(3223) return
[a] 0
[b] 10
[c] 5

[d] 1
Solution: (b)

D.) What is the output of the following program (in C) for input "Celestini Project"

```
#include "stdio.h"
int main()
{
   char arr[100];
   printf("%d", scanf("%s", arr));
   return 2;
}
```

[a] 0
[b] -1
[c] 1
[d] 2
Solution: (c)

E.) Which of the following options suggest the best approach to fix the high bias and high variance in a machine learning model? (Assume model has been trained on at least 1000 samples)
[a] To fix high bias, we can add more training samples; to fix high variance, we can reduce the number of training examples so it fits on them less
[b] To fix high bias, we can reduce our model's complexity; to fix high variance, we can increase our model's complexity
[c] To fix high bias, we can increase our model's complexity; to fix high variance, we can try reducing the number of features in the dataset
[d] To fix high bias, we can decrease the number of training samples; to fix high variance, we can increase the number of features in the dataset
Solution: (c)

F.) The major advantage(s) of prototyping over a Raspberry Pi over prototyping on a personal computer are
[a] cost
[b] faster processing speed
[c] small form factor
[d] low power consumption
Solution: (a, c, d)

G.) Which of the following statement(s) are correct?
[a] A machine learning model with higher accuracy will always indicate a better classifier.
[b] When we increase the complexity of a model, it will always decrease the test error.

[c] When we increase the complexity of a model, it will always decrease the train error.
Solution: (c)

H.) What is the output of the program (in C)?

```
 #include <stdio.h>
int main()
{
int celestini[6] = {6,5,4,3,2,1};
int *ptr = (int*)(&celestini+1);
printf("%d %d", *(celestini+1), *(ptr-1));
return 0;
}
```

[a] 5 1
[b] 4 3
[c] 6 4
[d] 5 3
Solution: (a)

I.) A poor binary classification model for detecting a **rare** cancer disease *always* predicts positive for presence of the disease. What can we infer about the model's performance?
[a] The model has high accuracy, maximum precision but low recall.
[b] The model has poor accuracy, poor precision but maximum recall.
[c] The model has poor accuracy, maximum precision and minimum recall.
[d] The model has maximum accuracy, maximum precision but minimum recall.
Solution: (b)

J.) Which of the following problems are best suited for a machine learning approach?
(i) Classifying numbers into primes and non-primes.
(ii) Detecting potential fraud in credit card charges.
(iii) Determining the time it would take a falling object to hit the ground.
(iv) Determining the optimal cycle for traffic lights in a busy intersection.
[a] (ii) and (iv)
[b] (i) and (ii)
[c] (i), (ii), and (iii).
[d] (iii)
Solution: (a)

2. Programming (10 points)

Given two sparse matrices A and B, perform multiply and convolution operation of the matrices in their sparse form itself. The result should consist of two sparse matrices, one obtained by multiplying the two input matrices, and the other obtained by convolution of the two matrices.

Recall that a sparse matrix is a matrix in which most of the elements are zero. Assume both the matrices are of size NxN. Assume the number of non-zero elements in A and B are m1 and m2 respectively. Note that other entries of matrices will be zero as matrices are sparse.
Note: You may use any data-structure to represent the sparse matrix. The solution approach should not use in-built libraries for the multiplication or convolution of matrices.

(i) Write code to solve the above problem in Python, Java or C++

<code inside the folder programming1>

(ii) What is the best time complexity of your solution (in terms of m1,m2,N)?

O(N*max(m1,m2))
What we do is that for every i,j in N*N, we multiply corresponding term of ith row of matrix 1 with corresponding term of jth row of matrix 2 so order is N*N*no of elements in each row no of elements in each row are roughly m1/N and m2/N respectively. Hence N*N*max(m1/N,m2/N)

To calculate the convolution, we went to each nonzero number and multiplied them if they shared the same row and column which again had the complexity max(m1,m2)

(iii) What is the best space complexity of your solution (in terms of m1,m2,N)?

We generated a matrix to store that non zero elements so the space complexity is max(m1,m2)

3. Programming II (10 points)

Write an efficient algorithm that searches for a value in an m x n matrix. This matrix has the following properties:

- Integers in each row are sorted in ascending from left to right.
- Integers in each column are sorted in ascending from top to bottom.

  For example,

  Consider the following matrix:

```
[
  [1,   4,  7, 11, 15],
  [2,   5,  8, 12, 19],
  [3,   6,  9, 16, 22],
  [10, 13, 14, 17, 24],
  [18, 21, 23, 26, 30]
]
```

Given target = 5, return true.
Given target = 20, return false.

(i) Write code to solve the above problem in Python, Java or C++.

<code inside folder programming2>

(ii) What is the best time complexity of your solution (in terms of m, n)?

O(m+n)

the maximum number of comparisons that would be done are m+n, because at every step we are either discarding one complete row or one complete column. And the total number of rows and columns are m+n

(iii) What is the best space complexity of your solution (in terms of m, n)?

O(1)

as the only extra variables we used are row and col

4. Problem Solving (20 points)

Please select either problem 4A or 4B and provide your solution in detail. You may solve both problems for extra credit though it is not required.

4A. Cryptosystem Identifier (select either 4A or 4B)

Cryptography is associated with the process of converting plain text into unintelligible text and vice versa. The goal of problem is to identify the cryptosystem used in encrypting a given cryptogram using Support Vector Machine (SVM) and Back propagation Neural Networks (BPNN). We consider that the cryptogram are derived using Simple substitution or Vigenere.
[a] Simple substitution (SS) ciphers work by replacing each plaintext character by another one character. To decode cipher text letters, one should use reverse substitution and change the letters back.
[b] Vigenere cipher is a kind of polyalphabetic substitution cipher. It is about replacing plaintext letters by other letters. Parties have to agree on a common shared keyword (which may also be a sentence), which is used during encryption algorithm.

Data generation approach: Create 50 cryptograms by Simple Substitution (Key size: 26) and 50 cryptograms by Vigenere cryptosystems (key size: 3). Each of the cryptograms should be of size 200 characters consisting of only upper case alphabets and white spaces (i.e. total 27 characters).
You can use the following links for encoding

- Vigenere: https://www.mathworks.com/matlabcentral/mlc-downloads/downloads/submissions/29443/versions/1/previews/VigenereDetails.html
- Simple substitution: https://in.mathworks.com/matlabcentral/fileexchange/31522-substitution-cipher-encoder-and-decoder

We are providing you with a dataset of ten plaintext, ten cryptograms by Vigenere, and ten cryptograms by simple substitution for testing your solution in the attachment (dataset_cryptosystem.doc)
Hint: You may consider using frequency pattern of the cryptograms for training the dataset.

(i) Write the solution for implementing Cryptosystem Identifier in MATLAB or Python. Give a brief description of what feature vectors you have used, how you designed the machine-learning model for SVM and BPNN, and what loss function did you use in each case.

Solution:
We observed frequency of characters in our cipher text and observed,

1. English Language has certain letter frequency. (wikipedia)
2. We observed that we arrange this character frequency then cipher text encrypted using simple substitution follows the character frequency of the english language if sorted.
3. This can be observed since if in the key the letter E is replaced by H then frequency of H would be same as that of E but it would appear at a different position if no rearrangement was done. If we arranged the frequencies in a descending order then vector where at each index we have frequency of occurence of a character in english alphabet. Then this vector would be same for cipher text and plain text.
4. However, this is not the case for ciphertext encrypted using vigener cipher. In that case if at one place E is replaced by H then it is not guaranteed that at some other place letter E is replaced by the same letter. Because if this is the case then encryption would be simple substitution only
5. Therefore using these feature vectors would help us learn parameters for vigenere and simple substitution cipher.

Several Observations have been provided in our jupyter notebook

For Neural Network we used input vector of 27 (1 bias + 26 features) and output to be one hot encoded vector where 0 [1 0] is for vigenere and 1 [0 1] is for simple substitution. We used a hidden layer of 256 nodes. Also we used softmax cross entropy loss function which measures the probability error in discrete classification tasks in which the classes are mutually exclusive and used Adam Optimizer with parameters (learning_rate=0.001, beta1=0.9, beta2=0.999, epsilon=1e-08)(adam).
For training we used 67 percent of training set and 33 percent as test set and achieved 99.85% accuracy on the test set.

For SVM we used radial basis function as the SVM kernel and other parameters (default values) mentioned here and obtained very good results. Since we obtained 99.8 percent accuracy on the test set. Therefore we do not need to tune hyperparameters. For Loss function sklearn library uses hinge loss, which is used for "maximum-margin" classification.

(ii) Compare the performance of the classifiers based on SVM and BPNN using test samples. Did you use a validation approach on the dataset? What performance metric did you use to compare the performance? Why is this a good metric?

Solution:
SVM and BPNN both had around 99.8 percent accuracy on the validation set.
We used Accuracy and F1 Score to compare performance. Since we had a balanced dataset (equal occurences of both the classes) accuracy is sufficient. Usually, If that is not the case accuracy alone is not sufficient therefore we used other metrics like F1 score to analyse our results.

(iii) Plot the performance of your system for SVM and BPNN by varying parameters in your model.
Solution:
Since we achieved about 99.8 percent on validation set and 100 percent accuracy on test set provided, hyperparameter tuning was not needed.

Results can be seen here

5. Solving socio-economic problems using technology (10 points)

Select one of the two problems below:
(i) Analytics and alerts on road safety using car mounted dashboard cameras

(ii) Analytics and alerts on air pollution in Delhi using vision and IoT sensors

Discuss in about 500-600 words how you would design a solution for the problem you selected above. Your solution approach needs to consider the following parts:
a) datasets or data acquisition for training
b) choice of machine learning algorithm to run online or offline
c) what platform can be used to run machine learning algorithm (for e.g. Raspberry Pi, smartphone, cloud)
d) sending alerts over the network via peer-to-peer methods or cloud architecture.

This question is open-ended so you need to outline the design choices you will make. Include an architecture diagram and how you would measure the performance of the system you design. What demo can you show and what key challenges do you expect. (Note: Additional credits on out-of the box feasible and interesting ideas)

Solution:

Real-time Detection using Deep Learning on
Embedded Platforms

Introduction

In this project, we will investigate real time vehicle detection, counting and distance estimation from video streams using wearable and embedded cameras. One possible application is the use of vehicle mounted cameras on Ola Cabs for Delhi-NCR pollution sensing. This is a project undertaken by Prof. Rijurekha Sen with a plan to fit pollution sensors with on-board cameras in collaboration with Ola. The collected data will be used for fine grained correlation analysis of air pollution with road traffic congestion levels. Detecting and counting vehicles from the on-board camera feed in real time will greatly reduce the store costs for raw video footage on resource constrained sensing equipment or communication costs for the raw video footage to be transmitted to a remote server.

Objectives

Our main objective is to make and deploy a model capable of detecting and classifying object of interests in a video. The envisioned computer vision software will process an incoming video feed and do vehicle counting and classification (into types like buses and other public transport, cabs, auto-rickshaws, personal cars, two-wheelers, heavy vehicles like trucks) in near real time. This information can be used in getting real-time information about traffic congestions, incidents and roadwork.

Datasets

1. PASCAL VOC: This dataset consists of realistic daily life images taken from Flickr. Intra-class variation is high and objects are viewed from multiple viewpoints. This dataset is composed of 20 object categories organized in four major groups.
2. MS-COCO: The COCO dataset is an excellent object detection dataset. It contains 91 common object categories with 82 of them having more than 5,000 labeled instances. Overall, the dataset has 2,500,000 labeled instances in 328,000 images.
3. Extensive video data on buses plying on various bus routes in Mumbai. This would

mimic a real deployment setting

Approach to the project:

1. We would focus primarily on recent architectures: SSD (Single Shot Multibox Detector), Faster R-CNN, R-FCN (Region-based Fully Convolutional Networks) and YOLO. While they were presented with a particular feature extractor (e.g., VGG, Resnet, etc), we would experiment with various feature extractors.
2. We would compare our results with standard benchmarks such as Imagenet and COCO with respect to accuracy.
3. In order to satisfy the constraints of the embedded platforms we could experiment with several architectural configurations, such as for Faster R-CNN and R-FCN, we can also choose the number of region proposals to be sent to the box classifier at test time. Typically, this number is 300 in both settings, but an easy way to save computation is to send fewer boxes potentially at the risk of reducing recall.
4. We would then be evaluating all the models based on accuracy/ latency/ energy or hardware metrics on Movidius Stick and NVIDIA Jetson TX2 Module.
5. An intuitive way to perform object detection in videos is to generate bounding box predictions for consecutive frames. We might need to drop frames in order to meet hardware constraints and improve latency.
6. Further Possibilities
   (a) We will look for techniques of data augmentation and transfer learning to compensate for the limited amount of training data.
   (b) We would look into using impression networks which might improve speed and performance simultaneously.

Challenges we might face:

1. Object detection from wearable cameras is challenging due to difference in view points based on the height of the person wearing the camera, his or her physical orientation and physical movements.
2. Processing high definition videos at powerful GPU servers might give the best performance in road traffic monitoring. But the poor broadband infrastructure in developing countries might prohibit real time streaming of HD videos from roads to servers. In-situ processing might mandate using mobile and embedded platforms, but their processor and battery constraints can conflict with heavy computation and low latency requirements. Our work would be to perform all the detection on embedded platforms like Intel Movidius sticks.
3. While accuracy of the inference task is an important metric to maximize, this might have trade-offs with other metrics on resource constrained embedded platforms. Is the latency of each inference too high to suit a real time mobile application while a user is interacting with it, or to suit a road traffic application to detect/prevent accidents? Is the trained deep-net model used in the inference too large to fit the embedded platform RAM? Does the inference task drain the mobile battery too fast? Our goal would be to see how different research communities are innovating to better handle these trade-offs.

Demonstration:

The most intuitive approach is to give a video as input and let the network detect and classify the objects of interest. But then we also need to place a check on the real time efficiency of the embedded system. So the best test would be to take the device on the road and check whether or not it classifies the traffic correctly.