COM2028/12/Semr2 18/19

UNIVERSITY OF SURREY ©

Faculty of Engineering and Physical Sciences

Department of Computer Science

Undergraduate Programmes in Computing
Undergraduate Programmes in Mathematics

Module COM2028: 15 credits

Introduction to Artificial Intelligence

FHEQ Level 5 Examination

Time allowed: Two hours

Semester 2 2018/19

Answer **ALL** questions

Approved calculators ARE permitted

Questions 1-5 carry 20, 15, 35, 15, and 15 marks respectively

Where appropriate the mark carried by an individual part of a question is indicated in square brackets [ ]

COM2028/12/Semr2 18/19

Solutions

1. This question is about *Image Processing.*

(a) Write a pseudo code that implements histogram equalisation using the cumulative histogram of a grey image as a mapping function. Explain when applying this algorithm to an input image, what the effect will be on the image.

[9 marks]

```
Given $f(x, y)$ representing the grey level intensity for pixel $(x, y)$ in an image
Compute a scaling factor, $a = 255/\text{number of pixels}$
Calculate the histogram of the grey image, histogram $[i]$, where $0 < i < 255$
$c[0] = a*\text{histogram}[0]$
for all remaining grey levels, $i$, do
    $c[i] = c[i-1] + a*\text{histogram}[i]$
end for
for all pixel coordinates, $x$ and $y$, do
    $g(x, y) = c[f(x, y)]$
end for
```

(5 marks)

The cumulative histogram of the image has such properties of having a steep slope ($a > 1$) at grey levels that occur frequently, and a gentle slope ($a < 1$) at unpopular grey levels. Thus when using the cumulative histogram as a mapping function, it will increase contrast for the most frequently occurring grey levels and reduce contrast in the less popular part of the grey level range. (4 marks)

(b) Given an image I, and three convolution kernels, A, B and C.

$$I = \begin{pmatrix} 2 & 1 & 1 & 2 \\ 0 & 1 & 0 & 3 \\ 1 & 3 & 2 & 5 \\ 3 & 0 & 2 & 5 \end{pmatrix} A = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 0 \end{pmatrix} B = \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix} C = \begin{pmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{pmatrix}$$

(i) Apply A to I, give the convolution result of I.

[5 marks]

(i)

$$I' = \begin{pmatrix} 1 & 1 \\ 2 & 2 \end{pmatrix}$$

(ii) Applying A, B and C to I, what will the resuling image look like when comparing with the orginal image I?

[6 marks]

Applying A will obtain a blurred I; applying B will obtain an image that highlights the horizontal edges in I; C will sharpen I.

Total marks: 20

2. This question is about *Regression Models and Their Related Concepts.*

(a) Suppose you are using Polynomial Regression. You plot the learning curves and notice that there is a large gap between the training error and the validation error. What is happening?

A. The model is overfitting the training set.

B. The model is overfitting the testing set.

C. The model is underfitting the training set.

D. The model is underfitting the testing set.

[4 marks]

A

(b) What could be the solutions to address the problem in (a)?

A. Increase the size of training data.

B. Try to regularize the model. For example, adding $l1$ or $l2$ penalty to the cost function.

C. Reduce the polynomial degree.

D. Increase the polynomial degree

[4 marks]

A, B, C

(c) Describe the gradient decent algorithm in at least three steps. Comment how the learning rate should be taken care of. [7 marks]

1) Initialize the search to start at initial set of learning parameters; 2) and let the gradient descent algorithm march downhill on the error function towards optimal solutions. Each iteration will update learning parameters to a better solution that yields slightly lower error than the previous iteration. 3) The direction to move in for each iteration is calculated using the partial derivatives of the error with regards to learning parameters.

Another way to describe Gradient Decent:

Define some cost function $J(w_1, w_2..., w_n)$ The objective is to minimise J

The algorithm will start with some $w_1, w_2..., w_n$ Then keep changing $w_1, w_2..., w_n$ to reduce J, until end up at a minimum. The change in each iteration is calculated using the partial derivatives of J with regards to $w_1, w_2..., w_n$.

(5 marks)

The learning Rate variable controls how large of a step the algorith takes downhill during each iteration. If it takes too large of a step, it may step over the minimum. However, if it takes small steps, it will require many iterations to arrive at the minimum. (2 marks)

Total marks: 15

3. This question is about *Neural Networks.*

(a) How many neurons do you need in the output layer if you want to classify email into spam or ham? What activation function should you use in the output layer? If instead you want to tackle MNIST (image dataset of handwriting digits from 0 to 9), how many neurons do you need in the output layer, using what activation function? [6 marks]

To classify email into spam or ham, we just need one neuron in the output layer of a neural network, indicating the probability that the email is spam. (2 marks) We could use the logistic activation function in the output layer when estimating a probability. (1 mark) For tackling MNIST, we need 10 neurons in the output layer, (2 marks) and use softmax activation function, which can handle multiple classes, outputting one probability per class. (1 mark)

(b) Consider a Convolutional Neural Network (CNN) composed of three convolutional layers, each with 3x3 kernels, a stride of 2, and SAME padding (which means the input image ought to have zero padding so that the output in convolution does not differ in size as input). The first layer outputs 100 feature maps, the middle layer outputs 200, and the last layer output 400 feature maps. The input image are RGB images (with red, blue, and green channels) of 400x400 pixels. What is the total number of parameters in the CNN? [8 marks]

Since its first concolutional layer has 3x3 kernels, and the input has threee channels (red, green, and blue), then each feature map has 3x3x3 weights, plus a bias term.That's 28 parameters per feature map. Since this first convoltutional layer has 100 feature maps, it has a total of 2,800 parameters. The second convolutional layer has 3x3 kernels, and its input is the set of 100 feature maps of the previous lauer, so each feature map has 3x3x100 = 900 weights, plus a bias term. Since it has 200 feature maps, this layer has 901x200 = 180,200 parameters. Finally, the third and the last convolutional layer also has 3x3 kernels, and its input is the set of 200 feature maps of the previous layers, so each feature map has 3x3x200=1,800 weights, plus a bias term. Since it has 400 feature maps, this layer has a total of 1,801x400 = 720,400 parameters. All in all, the CNN has 2,800+180,200+720,400=903,400 parameters.

(c) Based on the architecture described in question (b), continue the design of the neural network so that it can learn to classify the input images as either dinosaur, mammoth or blue whale. Add additional layers of network of any type as necessary. Justify your design (5 marks). Explain the training process (10 marks). Feel free to illustrate the network architecture by any drawing if that helps your explanation.

[15 marks]

SEE NEXT PAGE

Additional max pooling layer with further convolutional layers can be added. This will help the network to learn further high level features of the images. This can be followed by one or two layers of fully connected layers which connected by a softmax outputting three probabilities for three classes. (5 marks)

There are key elements in the training process. First define loss function (the cross entropy in this case) (2 marks); second, optimise parameters end-to-end by an optimisation algorithms such as Stochastic gradient descent (SGD). (2 marks)

During a training Loop: 1. input a sample a batch of data (1 mark) 2. Forward computation through the network graph, get loss (2 marks) 3. Backpropagation through applying the chain rule to compute the gradient of the loss function with respect to the inputs (2 marks) 4. Update the parameters using the gradient (1 mark)

(d) Identify at least one hyperparameter in your design and propose how to tune the hyperparameter(s) to improve the classification performance. [6 marks]

Hyperparameters: number of layers, size of kernels, 0 padding, stride size etc.

Hyperparameters can be tuned by performing validation. Split the training set in two: a slightly smaller training set, and a validation set. This validation set is essentially used as a fake test set to tune the hyper-parameters.

When the size of training data might be small, n-fold cross-validation can be used for hyperparameter tuning. Experimenting with different hyperparameter setting, iterate over different validation sets and averaging the performance across these. For example, in n-fold cross-validation, we would split the training data into n equal folds, use n-1 of them for training, and 1 for validation. We would then iterate over which fold is the validation fold, evaluate the performance, and finally average the performance across the different folds.

By the end of this procedure, we can decide which set of hyperparameter work best. We would then stick with these values and evaluate once on the actual test set.

Total marks: 35

4. This question is about *Unsupervised/semisupervised Machine Learning and Generative Models.*

Assume you are building a photo-hosting service which has the functionality of face recognition. Once a user uploads all their photos to the service, it automatically first conducts some unsupervised learning by clustering the same faces together. It then allows the user to tell it who these people are (i.e., to provide one label for each person), and it is able to name everyone in every photo. The labelled data can then be utilised for supervised learning about each person for face recognition from future incoming images.

(a) Design such learning algorithm and explain it it works. Assume we have already had a function to locate a face object, and the features of the face region are extracted. To simplify the problem, we also assume the number of people we are looking for in the photos is 6. [9 marks]

We can start with k-means clustering with k=6. The algorithm first assigns k random feature vectors representing k centres of k clusters. The distances between each data point (face object features) and each centre are calculated. (2 mark) The data point will be grouped to its closest centre. (1 mark) The centres will be updated based on their current group members. (1 marks) The whole process repeats until all members within one group stay the same. (2 marks) Once the user gives a label for any one member in a group, the label is them assigned to all members in the same group. So that all member feature vectors have their ground truth for a supervised training. (3 mark)

(b) List three methods that are capable of generating a synthesized face of any person. And propose at least one cost function for such work. (You can describe the cost function in text if not in a mathematics equation). [6 marks]

Genetic algorithms or other optimisation algorithms

or Autoencoder

or Generative Adversarial Networks (GANs).

(3 marks)

A cost function could be a L2 distance between the training sample and the generated sample. (3 marks)

Total marks: 15

5. This question is about *Gaussian Classifier and other related approaches.*

In a Gaussian classifier, the Gaussian probability density function for category w_i is given as below:

$$P(\mathbf{x} \mid w_i) = [(2\pi)^d \mid \sum_i |]^{-\frac{1}{2}} \exp[-\frac{1}{2}(\mathbf{x} - u_i)^T \sum_i^{-1} (\mathbf{x} - u_i)] \tag{1}$$

where

u_i — the mean vector of class w_i,

$\sum_i$ — the ith class covariance matrix.

$\mathbf{x}$ is the feature vector.

d is the dimension of $\mathbf{x}$.

u_i and $\sum_i$ are calculated from training samples belonging to category w_i

$u_i = \frac{1}{N_i} \sum_{j=1}^{N_i} \mathbf{x}_j, \mathbf{x}_j \in w_i$, where N_i is the number of training samples from class w_i.

The covariance matrix as

$$\sum_i = \frac{1}{N_i} \sum_{j=1}^{N_i} (\mathbf{x}_j - u_i)(\mathbf{x}_j - u_i)^T \tag{2}$$

A. What are the common computational steps and a key difference between a Gaussian classifier and a supervised k-means classifier. [6 marks]

B. How is Bayes Theorem applied in both a Gaussian Classifier and a Naïve Bayesian Classifier? [9 marks]

A. During the training processing, they both calculate u_i which is the mean (centre) for each class. The difference is that Gaussian classifier also considers the distribution (rather just the centre) of the training samples and uses both mean and covariance information of training samples to make its prediction on new instance.

B. In testing stage, both use Bayes Theorem in order to infer the posterior knowledge based on prior knowledge, i.e., given any testing datum x, we would like to know which classes or category it belongs to, and this can be done through:

$$P(w_i \mid \mathbf{x}) = \frac{P(w_i) * P(\mathbf{x} \mid w_i)}{P(\mathbf{x})} \tag{3}$$

where $P(\mathbf{x} \mid w_i)$ has been measured through training process. And often $P(w_i)$ and $P(\mathbf{x})$ are treated as scale factors.

In training phase, a Gaussian classifier measure the mean and covariance of each class based on training samples, so $P(\mathbf{x} \mid w_i)$ is known; while in a Naïve Bayesian Classifier, the training process takes the training data and their class labels to calculate the probabilities that each feature in the feature vector is associated with a particular class label based on all the training samples in each class, generating the probability about a certain class will contain a given feature, $p(\mathbf{x}^{(k)} \mid w_i)$, where k is the kth feature in the feature vector.

The feature probabilities need to be combined into a single probability for the entire piece of data $\mathbf{x}$.

$$P(\mathbf{x} \mid w_i) = p(\mathbf{x}^{(1)} \mid w_i) * p(\mathbf{x}^{(2)} \mid w_i) * \cdots \tag{4}$$

Total marks: 15

END OF PAPER

INTERNAL EXAMINER: DR. H.L.Tang
EXTERNAL EXAMINER: Prof M Roggenbach