COMP 2211 Final Exam (Part B) - Spring 2022 - HKUST

Date: May 27, 2022 (Friday)
Time Allowed: 1 hour 15 minutes, 2:15–3:30 pm
Instructions:

1. This is a closed-book, closed-notes examination.
2. In this part, there are **5** questions (The last one is a dummy problem) on **17** pages.
3. Write your answers in the space provided.
4. All programming codes in your answers must be written in the Python version as taught in the class.
5. For programming questions, unless otherwise stated, you are **NOT** allowed to define additional classes, helper functions and use global variables, nor any library functions not mentioned in the questions.

Student Name	SOLUTIONS AND MARKING SCHEME
Student ID	
Email Address	

For T.A. Use Only

Problem	Topic	Score
1	Convolutional Neural Network (CNN)	/ 19
2	Python Programming: Convolutional Neural Network	/ 13
3	Minimax and Alpha-Beta Pruning	/ 8
4	Ethics of Artificial Intelligence	/ 7
5	Dummy Question	/ 0
	Total	/ 47

Problem 1 [19 points] Convolutional Neural Network (CNN)

(a) [1.5 points] Suppose there is a convolutional layer in a CNN with the following parameters.

- Kernel size: 3×3
- Zero padding: 1-pixel border on each side
- Stride: 2-pixel in each direction

Compute the output feature map of the convolutional layer with the following input image and kernel. Assume the number of channels for both input and output images is 1. Express your answer as a 2D nested list. E.g. `[[2,2],[2,2]]`. If your calculated answer is a floating-point number, convert it to an integer using the floor function.

Input image:

2	4
1	3

Kernel:

1	6	7
2	5	8
3	4	9

Answer:

`[[73]]` for using the unflipped kernel OR
`[[27]]` for using the flipped kernel.

Marking scheme:

- Accepted answers:

  `[[73 (39, 43, 69)]], [[27 (61, 31, 57)]] (for flipped kernel)`

  -1 point if the format is wrong. e.g., missing bracket(s).

  If there are more than 1 value, the correct answer will be instead a 2x2 array extracted from the following arrays. There are two possible solutions:

  ```
  [A[1, 1], A[1, 2]], [A[2, 1], A[2, 2]]
  [A[1, 1], A[1, 3]], [A[3, 1], A[3, 3]] (stride 2 with extended zero padding)
  A = [[ 2, 16, 38, 28],
       [ 5, 27, 61, 53],
       [ 8, 31, 57, 60],
       [ 3, 13, 21, 27]]
  A = [[18, 44, 22, 12],
       [25, 73, 39, 17],
       [22, 69, 43, 10],
       [ 7, 27, 19,  3]]
  ```

  (this two array is generated by
  `scipy.signal.convolve2d(in1, in2, mode='full', boundary='fill', fillvalue=0)`)
  There are two possible arrays because traditionally, the former one is the "real" convolution. I.e. the kernel or image is flipped along both axes before the convolving operations. However, at the latter one (the originally intended answer), it's calculated WITHOUT the flipping. This operation is called the cross-correlation operation.

(b) [1.5 points] Give an example of a data augmentation technique that would be useful for classifying images of cats and dogs, but not for classifying handwritten digits. Briefly explain your answer.

Answer:

Flipping the image horizontally. Doing this to a dog/cat image would be reasonable, but not so for an image of a handwritten digit.

Marking scheme:

- 2 points for "rotate and reflections" as augmentation methods.
- 0 point for image processing methods.

(c) [4.5 points] Assume we have a CNN with the architecture as described in Table 1. Complete the table by filling in the input and output shape of each layer in the format: 'height x width x channel' (Note: A space is placed before and after the symbol 'x'). To illustrate that, the input shape of the 1st convolutional layer has been inserted for you.

Formula:
Convolution output size
= [(size of image dimension - size of kernel dimension + 2 × padding) / stride] + 1

If your calculated answer is a floating-point number, convert it to an integer using the floor function.

Layer (Size, Specifications)	**Input Shape**	**Output Shape**
7x7 conv, 64 kernels, stride 2, padding 3, with biases	224 x 224 x 3	112 x 112 x 64
3x3 max pooling, stride 2, padding 1	112 x 112 x 64	56 x 56 x 64
3x3 conv, 128 kernels, stride 2, with biases	56 x 56 x 64	27 x 27 x 128
3x3 conv, 256 kernels, stride 2, padding 1, with biases	27 x 27 x 128	14 x 14 x 256
3x3 conv, 512 kernels, stride 2, padding 1, with biases	14 x 14 x 256	7 x 7 x 512

Table 1: Architecture of a CNN

Marking scheme:

- If height/width is correct but wrong number of channels, 0.25 point deducted. But no points if height/width is incorrect at the first place.
- -1 point if the format is wrong, e.g. (c, h, w) instead of (h, w, c)

(d) [2 points] Assume we want to add a fully-connected layer at the back of the CNN as described in part (b). However, the output of the resulting network still contains a large number of parameters. State what we should do if we want to reduce the number of parameters and summarize the output.
Hint: We want to turn $N \times N \times 512$ output to $1 \times 1 \times 512$.
Remark: Convolution layer is not an acceptable answer.

Answer:
Max pooling or average pooling.

Marking scheme:

- 2 points for giving the correct answer.
- -0.5 point if type of pooling wasn't specified.
- -0.5 point if pooling is not mentioned. (e.g. taking average) As pooling is a standard component of neural networks.
- 1 point for resize.

(e) [2.5 points] Calculate the total number of parameters of the model? (i.e. for all the layers described in Table 1).

Answer:

$$\begin{aligned}\texttt{The total number of parameters} =& (7 \times 7 \times 3 \times 64 + 64) + \\ & (3 \times 3 \times 64 \times 128 + 128) + \\ & (3 \times 3 \times 128 \times 256 + 256) + \\ & (3 \times 3 \times 256 \times 512 + 512) \\ =& 9472 + 73856 + 295168 + 1180160 \\ =& 1558656\end{aligned}$$

Marking scheme:

- 2.5 point for giving the correct answer.

(f) Assume the model should classify images into 1000 distinct classes by

(i) [2.5 points] Appending the network shown in Table 2 to the end of the CNN network (i.e. the one described in Table 1) appended with the structure that you added in part (c).

Layer (Size, Specifications)	**Input Shape**	**Output Shape**
Flatten	1 x 1 x 512	512
Fully-connected (Flatten before feed-in)	512	1000

Table 2: Classification network

Calculate the total number of parameters in the whole network (i.e. the layers described in Table 1, your answer in part (d), and the layers described in Table 2).

Answer:

$$\begin{aligned}\texttt{The total number of parameters} &= 1558656 + (512 \times 1000 + 1000) \\ &= 1558656 + 513000 \\ &= 2071656\end{aligned}$$

Marking scheme:

- 2.5 point for giving the correct answer.

(ii) [2.5 points] Now, suppose we use the MLP described in Table 3 to classify the images instead of CNN.

Layer (Size, Specifications)	**Input Shape**	**Output Shape**
Flatten	224 x 224 x 3	150528
Fully-connected (Flatten before feed-in)	150528	1000

Table 3: Classification using MLP

Calculate the total number of parameters in the MLP (i.e. only those described in Table 3).

Answer:

$$\begin{aligned}\texttt{The total number of parameters} &= (150528 \times 1000 + 1000) \\ &= 150529000\end{aligned}$$

Marking scheme:

- 2.5 point for giving the correct answer.

(g) [2 points] State what you observe in part (f). Also, state the property of CNN, which leads to this difference.

Answer:

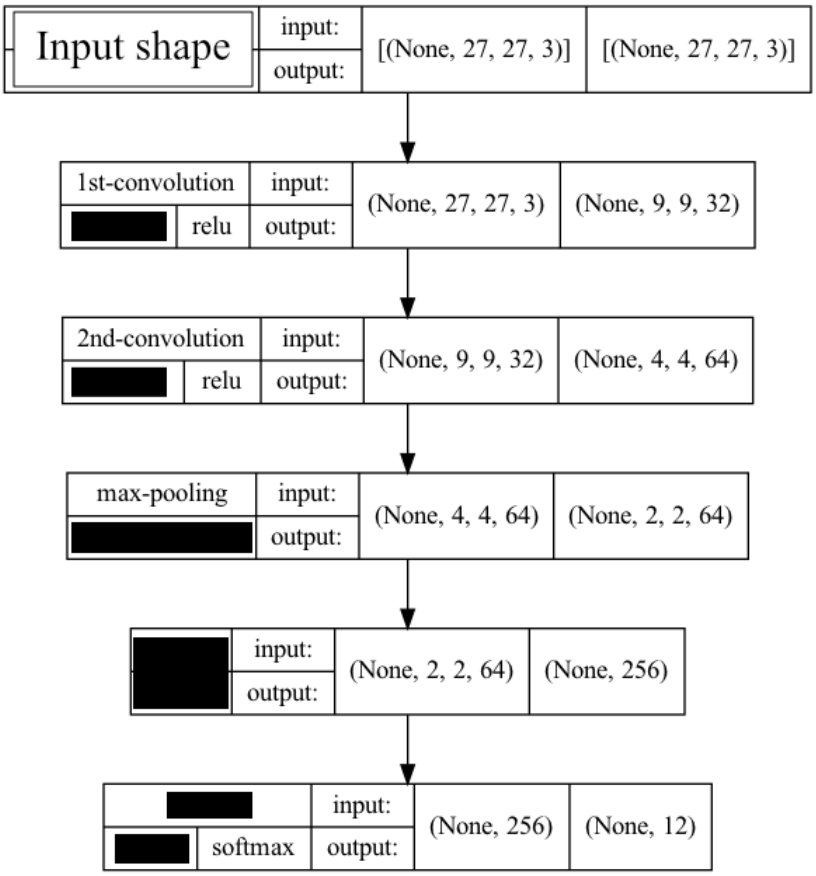
The number of parameters of CNN is significantly less than the number of parameters of pure MLP. This is because CNN uses shared parameters (kernels) to process the input instead of using 1 parameter for each individual input.

Marking scheme:

- 1 point for observations. If the observation is wrong, explanation is ignored.
- 1 point for explanation if mentioned "large scale extraction" but "feature extraction" only gains 0.5 point.
- 1 point for "sparse connection"
- 0 point for reducing output size. As fully-connected layer could also reduce the output size for the following fc layer, and yet the number of parameter is even more bloated.

Problem 2 [13 points] Python Programming: Convolutional Neural Network

Suppose the following CNN model learns human emotions from RGB images of faces and classifies into 12 emotion categories.



Assumptions: All layers have **no padding**. Both two convolutional layers have 3×3 **kernels**. The model is trained with Adam optimizer in default learning rate, and the loss function is categorical cross-entropy. You **don't need** to specify extra metrics like accuracy.

(a) [9 points] According to all the information given above, write the Python codes to construct and compile the model using Keras library. The following import statements are provided for you. Also, a reference of useful Keras classes and functions are given in the appendix.

```
from keras.models import Sequential
from keras.layers import Conv2D, MaxPooling2D
from keras.layers import Dense, Flatten
```

Answer:

```
model = Sequential()
model.add(Conv2D(filters=32, kernel_size=(3, 3), activation='relu',
                 strides=(3,3), input_shape=(32, 32, 1)))
model.add(Conv2D(filters=64, kernel_size=(3, 3), activation='relu', stride=(2,2)))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Flatten())
model.add(Dense(units=12, activation='softmax'))
model.compile(optimizer='adam', loss='categorical_crossentropy')
```

Marking scheme:

- Apply the following rules until the score is 0.
  - -1 for each of Pooling, Flatten, Dense, Compile lines if it is missing or has any wrong parameter
  - -1 for each of the wrong or missing parameters in a Conv2D layer. This includes any parameter that is good by default but you break it by setting another absolutely wrong value.
  - -1 if the input shape is not specified in any manner
  - -1 for syntax errors like messed up brackets, the layers are not actually added to a model, etc. Small typos on keywords, etc. are not penalized

(b) [4 points] Now instead of classification, a student wants to predict a single continuous real value from an input picture: how positive (or negative) the emotion is, ranging from 0 to 1. Is CNN in general able to do that? If yes, derive a model to complete such a task from part (a) model. Point out which parts you will change when building and compiling the model (if any) and explain how you will change it. (You don't have to write codes, but state clearly your ideas). If no, also explain why.

Answer:

Yes, CNN is able to do that in general, and the following parts need to be changed.

- Change the Dense layer to units=1.
- Change the Dense layer activation to any other than softmax, e.g., relu or sigmoid.
- Change the loss to any regression loss, e.g., MSE, MAE.

Marking scheme:

- 1 point for stating CNN is able to do that in general.
- 1 point for changing the output Dense unit to 1
- 1 point for changing output activation to anything not related to probability and range containing [0,1] (sigmoid, linear, relu, etc.)
- 1 point for changing the loss function to any real value comparison (subtraction), e.g. Mean Squared Error, Mean Absolute Error, or other similar self-defined losses.
- -0.5 point for each change if the student doesn't specify to what it changes or changes it to a wrong value.

Appendix:

Below are some Keras documentation for your reference. Some irrelevant parameters are omitted for conciseness.

Sequential class

```
tf.keras.Sequential(layers=None, name=None)
```

Sequential groups a linear stack of layers into a `tf.keras.Model`

- **add Method**

  `Sequential.add(layer)`

  - **layer**: layer instance
- **compile Method**

  `Model.compile(optimizer="rmsprop", loss=None)`

  - **optimizer**: String (name of optimizer) or optimizer instance.
  - **loss**: Loss function.

Conv2D class

```
tf.keras.layers.Conv2D(
    filters, kernel_size, strides=(1, 1), padding="valid", activation=None,
)
```

2D convolution layer (e.g. spatial convolution over images). When using this layer as the first layer in a model, provide the keyword argument `input_shape` (tuple of integers, does not include the sample axis), e.g. `input_shape=(128,128,3)` for 128x128 RGB pictures in `data_format="channels_last"`.

- **filters**: Integer, the dimensionality of the output space (i.e. the number of output filters in the convolution).
- **kernel_size**: An integer or tuple/list of 2 integers, specifying the height and width of the 2D convolution window. Can be a single integer to specify the same value for all spatial dimensions.
- **strides**: An integer or tuple/list of 2 integers, specifying the strides of the convolution along the height and width. Can be a single integer to specify the same value for all spatial dimensions. Specifying any stride value != 1 is incompatible with specifying any `dilation_rate` value != 1.
- **padding**: one of "`valid`" or "`same`" (case-insensitive). "`valid`" means no padding. "`same`" results in padding with zeros evenly to the left/right or up/down of the input. When `padding="same"` and `strides=1`, the output has the same size as the input.
- **activation**: Activation function to use. If you don't specify anything, no activation is applied.

MaxPooling2D class

```
tf.keras.layers.MaxPooling2D(
    pool_size=(2, 2), strides=None, padding="valid",
)
```

Max pooling operation for 2D spatial data.

- **pool_size**: integer or tuple of 2 integers, window size over which to take the maximum. `(2, 2)` will take the max value over a 2x2 pooling window. If only one integer is specified, the same window length will be used for both dimensions.
- **strides**: Integer, tuple of 2 integers, or None. Strides values. Specifies how far the pooling window moves for each pooling step. If None, it will default to `pool_size`.
- **padding**: One of "`valid`" or "`same`" (case-insensitive). "`valid`" means no padding. "`same`" results in padding evenly to the left/right or up/down of the input such that output has the same height/width dimension as the input.

Flatten class

```
tf.keras.layers.Flatten()
```

Flattens the input.

Dense class

```
tf.keras.layers.Dense(
    units, activation=None,
)
```

Regular densely-connected NN layer.

- **units**: Positive integer, dimensionality of the output space.
- **activation**: Activation function to use. If you don't specify anything, no activation is applied.

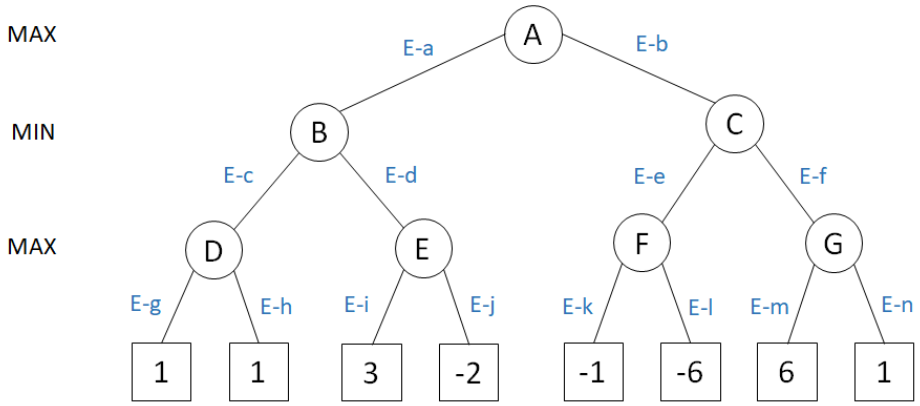
Common activation functions (in shorthand strings): "`relu`", "`sigmoid`", "`softmax`"

Common loss functions (in shorthand strings):
"`categorical_crossentropy`",
"`sparse_categorical_crossentropy`",
"`mean_squared_error`" (same as "`MSE`"),
"`mean_absolute_error`" (same as "`MAE`")

Common optimizer (in shorthand strings): "`adam`"

Problem 3 [8 points] Minimax and Alpha-Beta Pruning

Two players, MAX and MIN, are playing a game that can be represented by a tree, as shown below.



(a) [3.5 points] Complete the following table by estimating the minimax value of each non-terminal node.

Answer:

A	1
B	1
C	-1
D	1
E	3
F	-1
G	6

Marking scheme:

- 0.5 point for each correct answer. 3.5 points in total.

(b) [0.5 point] State the proper move of the maximizer by writing down one of the root's outgoing edges (i.e. E-a or E-b).
Note: The root node of the given tree is A.

Answer:
E-a

Marking scheme:

- 0.5 point for the correct answer.

(c) [1.5 points] State whether minimax-based AI will choose to make a move which will result in a slower victory. Explain your answer.

Answer:

Yes, minimax-based AI may choose to make a move, resulting in a slower victory. Since we may have two moves with the same maximum minimax value, it picks the one in slower victory.

Marking scheme:

- 1 point for stating minimax-based AI will choose to make a move which will result in a slower victory.
- 0.5 point for giving the proper explanation.

(d) [2.5 points] Suppose we now apply alpha-beta pruning on the game tree. Indicate the edge(s) that would be pruned (eliminated from consideration) by writing down the edge labels (i.e. E-a, E-b, E-c, ..., E-n). You may assume that the branches are explored from left to right.

Answer:

E-j and E-f would be pruned.

Marking scheme:

- If E-j appears: +1 point.
- If E-f appears: +1.5 point.
- If E-m or E-n appears: +0 point.
- If other label appears: then d) become 0 points, no matter E-j and E-f appears or not.

Problem 4 [7 points] Ethics of Artificial Intelligence

This question consists of five sub-questions, four of them are multiple-choice questions, and one is a short question. Choose the **BEST ANSWER** among the given choices for each multiple-choice question and put your answer in the given table, while for the short question, answer it in a few sentences.

(a) [1 point] Ethics in artificial intelligence is

(A) Something that is not an issue.

(B) Something that somebody else will do in the future.

(C) Something that we need to apply today.

(D) Something that is entirely solved in current AI systems.

(b) [1 point] One approach that helps developers avoid unintentionally creating bias in AI systems is

(A) Using a wide variety of appropriately diverse data for training.

(B) Using highly specific training data from a narrow range.

(C) Not using any training data.

(D) None of the above

(c) [1 point] What are some of the ethical concerns around artificial intelligence?

I. Racial, gender or other types of bias.

II. Loss of jobs due to AI replacing workers performing repetitive tasks.

III. Concern about the trustworthiness of decision-making supported by AI systems.

IV. Privacy, for example, as human faces are photographed and recognized in public spaces.

(A) I and II only

(B) I, II, and IV only

(C) All of the above

(D) None of the above

(d) [1 point] What is a significantly way in which developers of AI systems can guard against introducing bias?

(A) Using only examples from their own environment as training data.

(B) Providing effective training data and performing regular tests and audits.

(C) Using less varied AI systems and datasets.

(D) Using government approved algorithms.

Question	Answer
(a)	C
(b)	A
(c)	C
(d)	B

Marking scheme:

- 1 point for each correct answer. 4 points in total.

(e) [3 points] State THREE ethical issues involved with the introduction of autonomous vehicles.

Answer:

- Who is to blame in an accident?
- In an emergency situation who should the car prioritize?
- Increase in use of cars is bad for the environment.
- Cost of the cars.

Marking scheme:

- 1 point for giving each correct ethical issue. 3 points in total.

Problem 5 [0 points] Dummy Question

Are you sure you have finished all the questions?

---------- END OF PAPER ----------