Venue	Online
Student Number	20169321
Family Name	Chitete
First Name	Tanaka

Curtin University

School of Electrical Engineering, Computing and Mathematical Sciences

Mid-Semester Test (Open Book)

11th of April 2022

COMP3010 Machine Learning

Examination Duration

75 Minutes

Total Marks

100

Supplied by the University

None

Supplied by the Student

Materials

Any text books, lecture notes or written materials are permitted.

Internet access is not allowed except for access to Blackboard and IRIS.

Instructions to Students

You can type with any editor or write by hand. In either way, you need to save or scan the document of your answers as a pdf file and submit the pdf file through Blackboard. Both the Word and the PDF files of the assessment paper are provided. If you write by hand, you should print the pdf file and write your answers clearly and neatly, and make sure the scanned pdf file is clear. Messy handwriting, or poor quality of the scanned pdf file may get marks deducted or get zero marks if the pdf file is not readable. You can write on extra pages if extra space is needed. Even if the system is still open for submission after the due time, late submission will get zero marks. Multiple attempts are allowed but only the last attempt will be marked, which means, if the last attempt is late, it will get zero marks. **During the assessment period, internet access is not allowed except for access to Blackboard. Searching for answers from websites or sharing answers with others are contract cheating and are not permitted. Also, you need to sign the student declaration form on the next page.**

Curtin University

Faculty of Science and Engineering

Unit Code: COMP3010

Unit Name: Machine Learning

Unit Coordinator: Senjian An

Mid-Semester Test

(Assessment Title)

STUDENT DECLARATION:

I declare that this assessment item is my own unassisted work, except where acknowledged, and it has not been submitted in any form for assessment or academic credit elsewhere.

I certify that I have read and understood Curtin University policies on Plagiarism and Copyright and declare that this assessment item complies with these policies.

I certify that I have adhered to the time duration limit prescribed for the completion of this assessment item.

I recognise that should this declaration be found to be false, disciplinary action could be taken and penalties imposed in accordance with Curtin University policy.

Full Name of Student: Tanaka Chitete

Student ID Number: 20169321

Signature: TANAKA CHITETE

Date: 11/04/2022

Question 1 – Linear Models and Multilayer Perception (20 marks)

Consider a house price prediction problem based on three different type of features: the number of bed rooms, the land size and the location (x,y).

- With this example, describe the limitations of linear models.

[8 marks]

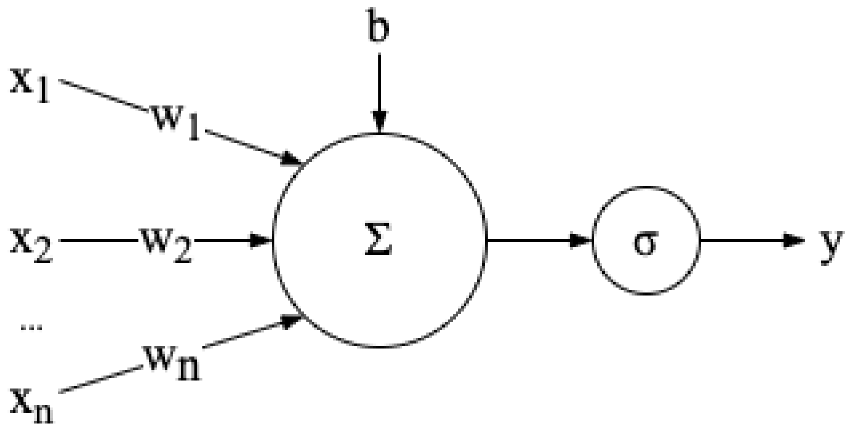
Linear models lead us to make the assumption of *monotonicity*–any increase in our feature (number of bedrooms, land size, and location) must either always cause an increase in our model's output (if the corresponding weight is positive), or always cause a decrease in our model's output (if the corresponding weight is negative). Sometimes, this is a reasonable assumption. For example, Since we are predicting house prices, we might reasonably imagine that all else equal, a house with a bigger land size income would always be more likely to be worth more than a house with a smaller land size. While monotonic, this relationship likely is not linearly associated with the probability of house price. An increase in land size 100 to 200 thousand likely corresponds to a bigger increase in house price than an increase from 1000 to 1100. One way to handle this might be to pre-process our data such that linearity becomes more plausible, say, by using the logarithm of house price as our feature.

Another limitation is limited capacity–we can only separate linearly separable data. A dataset is linearly separable if for each class there is a hyper plane to separate itself from the the others.

- With this example, define a neuron and describe how neurons can be organised to define a nonlinear model which can approximate any nonlinear mapping between the input (i.e., the features) and the output (i.e., the house price).

[12 marks]

A neuron is a singular unit which (i) takes some set of inputs; (ii) generates a corresponding scalar output; and (iii) has a set of associated parameters that can be updated to optimise some objective function of interest. An illustration of such a neuron is given in the below figure.



Neurons can be organised into input and output layers with several intermediate hidden layers where the neurons form wide layers (layers with many neurons) or are organised into several layers (constituting a deep neural network).

Answer:

Question 2 – Implementation of Neural Networks (20 Marks)

Consider an image classification problem with 10 classes (e.g. the 10 digits), and the input image size is 30x30.

Question 2.1: Multilayer Perception (MLP)

Suppose you are implementing an MLP network with three hidden layers where the first and the third hidden layers share both their weights and biases, and a linear output layer. Define the MLP model with nn.sequential().

[10 marks]

```
shared = nn.Linear(900, 900)
net = nn.Sequential(nn.Flatten(),
                    shared, nn.ReLU(),
                    nn.Linear(900, 900), nn.ReLU(),
                    shared, nn.ReLU(),
                    nn.Linear(900, 10),
                    nn.Softmax())
```

Question 2.2: Convolutional Neural Network (CNN)

Suppose you are implementing a CNN with one convolutional hidden layer with a 3x3 convolution kernel, and a linear output layer. Define the CNN model with nn.sequential().

[10 marks]

```
net = nn.Sequential(nn.Conv2D(10, kernel_size=(3, 3), strides=(1, 1)
                    nn.Linear(900, 10))
```

Answer:

Question 3 – Convolution and max Pooling (20 Marks)

Given below is an input matrix of shape 7 X 7

1	2	4	1	4	0	1
0	0	1	6	1	5	5
1	4	4	5	1	4	1
4	1	5	1	6	5	0
1	0	6	5	1	1	8
2	3	1	8	5	8	1
0	9	1	2	3	1	4

Figure 1. Input matrix for convolution and max pooling.

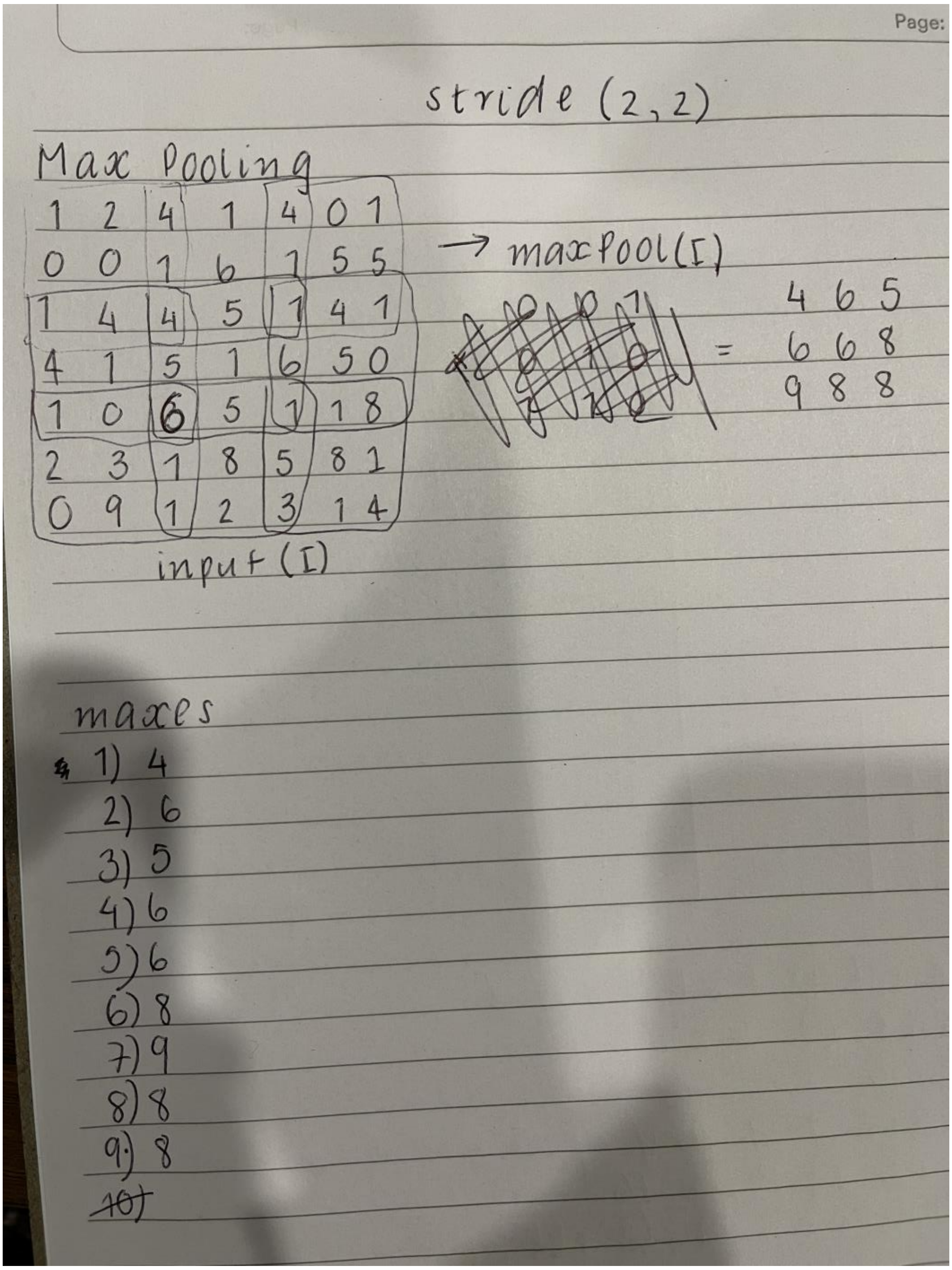
Consider a 3x3 convolution kernel in Figure 2.

0	0	1
0	1	0
1	1	0

Figure 2. A 3x3 convolution kernel.

Question 3.1. What will be the output after applying a max pooling of size 3 X 3 with a stride of 2?

[10 marks]



Question 3.2. What will be the output after convolution with a stride of 3 and the kernel in Figure 2?

[10 marks]

Convolution

$$\begin{matrix} 1 & 2 & 4 & 1 & 4 & 0 & 1 \\ 0 & 0 & 1 & 6 & 1 & 5 & 5 \\ 1 & 4 & 4 & 5 & 1 & 4 & 1 \\ 4 & 1 & 5 & 1 & 6 & 5 & 0 \\ 1 & 0 & 6 & 5 & 1 & 1 & 8 \\ 2 & 3 & 1 & 8 & 5 & 8 & 1 \\ 0 & 9 & 1 & 2 & 3 & 1 & 4 \end{matrix} \quad \times \quad \begin{matrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \end{matrix} \quad = \quad \begin{matrix} 9 & 7 \\ 10 & 19 \end{matrix}$$

$\lfloor (7 - 3 + 0 + 3)/3 \rfloor \times \lfloor (7 - 3 + 0 + 3)/3 \rfloor$

$= \lfloor 7/3 \rfloor \times \lfloor 7/3 \rfloor$

$= 2 \times 2$

$0,0) = (4 \times 1) + (1 \times 1) + (4 \times 1)$
$= 4 + 1 + 4$
$= 9$

$0,1) = (1 \times 1) + (5 \times 1) + (1 \times 1)$
$= 1 + 5 + 1$
$= 7$

$1,0) = (5 \times 1) + (2 \times 1) + (3 \times 1)$
$= 5 + 2 + 3$
$= 10$

$1,1) = (5 \times 1) + (1 \times 1) + (8 \times 1) + (5 \times 1)$
$= 5 + 1 + 8 + 5$
$= 19$

Answer:

Question 4 – Regularisation (15 Marks)

List out two regularisation methods which are commonly used to prevent overfitting in the training of neural networks, and briefly explain how these methods are used in training of neural networks.

Answer:

One regularisation method is *weight decay* (commonly called L_2 regularisation). With this method, we compute the norm of the weight vector and add it as a penalty term in the loss function. Therefore, rather than minimising the prediction loss on the training labels, we opt to minimise the sum of the prediction loss and the penalty term. Now, if the norm of the weight vector is too large, our learning algorithm might focus on minimising the weight norm rather than the training error, reducing the likelihood of overfitting.

Another such method is *dropout*. With this method, we inject noise while computing each hidden layer during training by literally dropping out some neurons. Throughout training, on each iteration, standard dropout consists of zeroing out some fraction of the nodes in each layer before calculating the subsequent layer. Moreover, we ensure to dropout neurons in an unbiased manner so that the expected value of each layer–while fixing the others–equals the value it would have taken without dropout.

Question 5 – Cross-validation (15 Marks)

Briefly describe 1) what is cross-validation? 2) What is it used for? and 3) when cross-validation should be used?

Answer:

1. Cross-validation–rather, *K*-Fold cross-validation–is the process wherein the original training data is split into *K* non-overlapping subsets.
2. It is used to partition the training data into batches prior to executing the training loop.
3. It should be used when the training data is scarce.

Question 6 – Universal Approximation (10 Marks)

Briefly explain 1) what does " A neural network with one hidden layer is a universal approximator" mean?, and 2) why this is important?

Answer:

1. It means that it can approximate any non-linear function.
2. This is important because with the right amount of nodes (which may be excessive), and the right set of weights, we can model any function. However, it should be noted that learning said function can be quite difficult.

END OF ASSESSMENT