1(a)

[2] (i) Find $\dfrac{\partial YS(i,j)}{\partial C}$ for all relevant i and j.

$$\underset{X}{\begin{bmatrix} 0 & 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 & 9 \\ 0 & 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 & 9 \\ 0 & 1 & 2 & 3 & 4 \end{bmatrix}} \xrightarrow{C = \begin{bmatrix} a & b \\ c & d \end{bmatrix}} \underset{Y}{\begin{bmatrix} 2 & ? & 5 & ? \\ ? & ? & ? & ? \\ ? & ? & ? & 4 \\ 4 & ? & ? & ? \end{bmatrix}} \xrightarrow{\text{Max Pooling}} \underset{YS}{\begin{bmatrix} 2 & 5 \\ 4 & 4 \end{bmatrix}}$$

[2] (ii) Find $\dfrac{\partial YS(i,j)}{\partial X}$ for all relevant i and j.

(i) $\dfrac{\partial YS(1,1)}{\partial C} = \begin{bmatrix} & 1 \\ 5 & 6 \end{bmatrix}$ $\qquad \dfrac{\partial YS(1,2)}{\partial C} = \begin{bmatrix} 2 & 3 \\ 7 & 8 \end{bmatrix}$

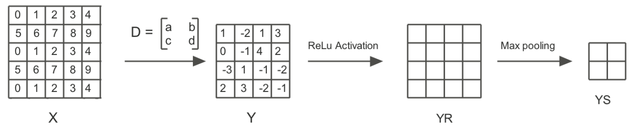
$\dfrac{\partial YS(2,1)}{\partial C} = \begin{bmatrix} 5 & 6 \\ & 1 \end{bmatrix}$ $\qquad \dfrac{\partial YS(2,2)}{\partial C} = \begin{bmatrix} 3 & 4 \\ 8 & 9 \end{bmatrix}$

(ii) $\dfrac{\partial YS(1,1)}{\partial X} = \begin{bmatrix} a & b & & & \\ c & d & & & \\ & & & & \\ & & & & \\ & & & & \end{bmatrix}$ $\qquad \dfrac{\partial YS(1,2)}{\partial X} = \begin{bmatrix} & & a & b & \\ & & c & d & \\ & & & & \\ & & & & \\ & & & & \end{bmatrix}$

$\dfrac{\partial YS(2,1)}{\partial X} = \begin{bmatrix} & & & & \\ & & & & \\ & & & & \\ a & b & & & \\ c & d & & & \end{bmatrix}$ $\qquad \dfrac{\partial YS(2,2)}{\partial X} = \begin{bmatrix} & & & & \\ & & & & \\ & & & a & b \\ & & & c & d \\ & & & & \end{bmatrix}$

1(b)

[2] (i) Find $\frac{\partial YS(i,j)}{\partial D}$ for all relevant i and j.



[2] (ii) Find $\frac{\partial YS(i,j)}{\partial X}$ for all relevant i and j.

$YR =$

1		1	3
		4	2
	1		
2	3		

$YS =$

1	4
3	0

(i) $\frac{\partial YS(1,1)}{\partial D} =$

	1
5	6

$\frac{\partial YS(1,2)}{\partial D} =$

7	8
2	3

$\frac{\partial YS(2,1)}{\partial D} =$

6	7
1	2

$\frac{\partial YS(2,2)}{\partial D} =$

(ii) $\frac{\partial YS(1,1)}{\partial X} =$

a	b			
c	d			

$\frac{\partial YS(1,2)}{\partial X} =$

		a	b	
		c	d	

$\frac{\partial YS(2,1)}{\partial X} =$

	a	b		
	c	d		

$\frac{\partial YS(2,2)}{\partial X} =$

[4] **1(c)**
Consider the part of a Mini-CNN shown below:

M(7 by 7), C(2 by 2), $C(i,j) = i+j$, stride 1, Y, 2 by 2 Maxpooling, YS, D(2 by 2), $D(i,j) = i+j+(-1)^i$, stride 1, YY

0	0	0	0	0	0	0
1	1	1	1	1	1	1
0	0	0	0	0	0	0
1	0	1	0	1	0	1
0	0	0	0	0	0	0
0	1	0	1	0	1	0
0	1	0	1	0	0	1

M

If M is given by the pixels above, find $\frac{\partial yy_{2,2}}{\partial M}$. No need to simplify your answer.

$Y: \frac{7-2}{1}+1=6 \qquad YY: \frac{3-2}{1}+1=2$

$M(7\times7) \xrightarrow[C=\begin{bmatrix}2&3\\3&4\end{bmatrix}]{} Y(6\times6) \xrightarrow[\text{maxpool }(2\times2)]{} YS(3\times3) \xrightarrow[D=\begin{bmatrix}1&2\\4&5\end{bmatrix}]{} YY(2\times2)$

$M =$

1	1	1	1	1	1	1
1		1		1		1
	1		1		1	
	1		1			1

$Y =$

7	7	7	7	7	7
5	5	5	5	5	5
3	4	3	4	3	4
2	3	2	3	2	3
4	3	4	3	4	3
7	5	7	5	3	6

$YS =$

7	7	7
4	4	4
7	7	6

$\frac{\partial YY_{22}}{\partial YS} =$

	1	2
	4	5

$\frac{\partial YY_{22}}{\partial M} = 1\times$

			2	3		
			3	4		

$+ 2\times$

					2	3
					3	4

$+ 4\times$

		2	3			
		3	4			

$+ 5\times$

					2	3
					3	4

[4.7] **1(d)**
Find the total number of parameters in the following CNN:

M_{1-3}(74 by 74), C_{1-2}(5 by 5), stride 3, Y, YR, 2 by 2 Maxpooling, YS, z_{1-20}, zr_{1-20}, zz

$Y: \frac{74-5}{3} + 1 = 24$ $YR: 24$ $YS: 24/2 = 12$

$$\underset{(3)}{M(74\times74)} \xrightarrow{C} \underset{(2)}{Y(24\times24)} \longrightarrow \underset{(2)}{YR(24\times24)} \longrightarrow \underset{(2)}{YS(12\times12)}$$

$\Downarrow$ $w = 3\times2\times5\times5 = 150$, $b = 2$ $\Downarrow$ 0 $\Downarrow$ 0

$$\underset{(2)}{YS(12\times12)} \longrightarrow Z_{1-20} \longrightarrow Zr_{1-20} \longrightarrow zz$$

$\Downarrow$ $w = 2\times12\times12\times20 = 5760$, $b = 20$ $\Downarrow$ 0 $\Downarrow$ $w = 20$, $b = 1$

$\Rightarrow$ Total $= 150 + 2 + 5760 + 20 + 20 + 1 = 5953$

2(a)

I am performing single time series prediction. I have the input data in an array of shape 18000 by 1. My plan is to give 36 numbers in one window and get the SimpleRNN to to predict the next 2 numbers in the series. I am planing to use the reshape method. I want to use a window size of 12 and give 3 inputs per time slot. I wish to train 2 outputs at the end of the window.

[2] (i) Find the shape of the input tensor. (No part marks.)

[2] (ii) Find the shape of the target array. (No part marks.)

(i) $18000/36 - 1 = 499$ (# of samples)

(499, 12, 3)

(ii) (499, 2)

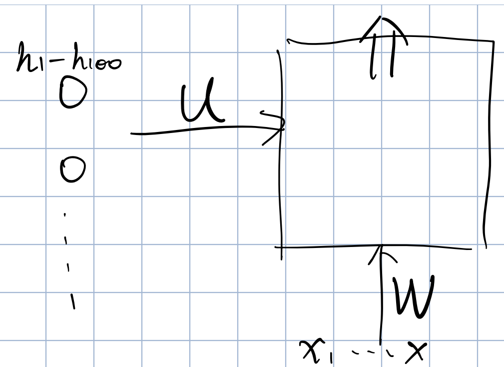
[1] **2(b)**

Write the model part of of the SimpleRNN TensorFlow code to solve my problem above.

```
model = sequential([
    layers.SimpleRNN(100, input_shape = (12,3), activation= 'relu')
    layers. Dense (2, activation = 'linear')
  ])
```

[4.7] **2(c)**

I am using a window size of 5 and giving 3 inputs per time slot to a SimpleRNN TensorFlow model in its default setting. I want to train 2 outputs in each window. If I am using 100 neurons in the first layer of my model, find the total number of parameters in the ANN inside my model.



$$\left.\begin{array}{l} \left.\begin{array}{l} U: 100 \times 100 = 10000 \\ W: 3 \times 100 = 300 \\ \text{bias}: 100 \end{array}\right\} 10400 \\ \left.\begin{array}{l} WW: 100 \times 2 = 200 \\ \text{bias}: 2 \end{array}\right\} 202 \end{array}\right\} 10602$$

[5] **2(d)**
Let $y(i+2) = ay(i+1) + by(i) + 1,\ i = 1, 2, 3, \ldots.$

Find $\dfrac{\partial y(4)}{\partial a}$. Make the necessary assumptions (as little as possible) about some of the initial values. Do not make any unnecessary assumptions.

Hint: The easiest method is to start by putting $i = 2$ and writing out the definition of $y(4)$.

$y(4) = ay(3) + by(2) + 1$

$y(3) = ay(2) + by(1) + 1$

$y(2) = ay(1) + by(0) + 1$

$y(1) = ay(0) + by(-1) + 1$

since $y(i)$ is only defined for $i = 1, 2, 3 \ldots$

$\Rightarrow$ there is no $y(0)$ and $y(-1)$

$\Rightarrow$ cannot compute $y(1)$ and $y(2)$ using equation

$\Rightarrow$ assume $y(1) = 1 \quad y(2) = 1$

$\Rightarrow y(3) = a+b+1 \quad , \ y(4) = a(a+b+1) + b + 1$

$\Rightarrow \dfrac{\partial y(4)}{\partial a} = 2a + b + 1$

[1] **2(e)**
Python list f has the English alphabets in the natural order. The list r has the same alphabets but in the reverse order. What would be output from the following line?

```
print(f[f.index('x') - 20])
```

f.index('x') = 23

f[3] = 'd' $\Rightarrow$ d

	0	1	2	3	4
0	a	b	c	d	e
1	f	g	h	i	j
2	k	l	m	n	o
3	p	q	r	s	t
4	u	v	w	x	y
5	z				

[1] **2(f)**

Python list x has 1000 elements. The last 26 of them were English alphabets in the natural order. What would be the output of the following script?

```
import numpy as np
y=np.reshape(x,(100,10))
print(y[99,6])
```

y[99, :] = ['q', 'r', ... 'z']

y[99,6] = 'w' $\Rightarrow$ w.

4.

(a) Suppose X = ones$(4,4)$ and $Y(i,j) = (-1)^j$ones$(4,4)$. If I require $X \bullet W = 8$ and $\cancel{X} \bullet W = 0$, guess and write down one possible solution for W.

Y

$$X = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \qquad Y = \begin{bmatrix} -1 & -1 & -1 & -1 \\ 1 & 1 & 1 & 1 \\ -1 & -1 & -1 & -1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

$$\Rightarrow W = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ -1 & -1 & -1 & -1 \end{bmatrix}$$

(b) In a 60 by 60 featuremap the following filters and strides were used to obtain another featuremap. In each case, find the dimension of the resulting image.

(i) 20 by 20 with stride 1 (ii) 20 by 20 with stride 2. (iii) 10 by 10 with stride 1. (iv) 10 by 10 with stride 2.

bi. $(60-20)/1+1 = 41$
bii. $(60-20)/2+1 = 21$
biii. $(60-10)/1+1 = 51$
biv. $(60-10)/2+1 = 26$

(c) In a 40×40 featuremap, the following subsampling filters ware used to create reduced featuremaps. In ecah case, find the dimension of the resulting featuremap.

(i) 2 by 2 (ii) 4 by 4

ci. $40/2 = 20 \Rightarrow 20 \times 20$
cii. $40/4 = 10 \Rightarrow 10 \times 10$

(d) I want to go from a size 40 map to a size 18 map using convolution alone. Find all possible filter sizes and the corresponding strides. Do not use a stride longer than the filter size. Zero padding is not allowed.

$(40-a)/b+1=18 \quad (a \geq b)$

$b=1 \Rightarrow a=23 \quad \Rightarrow$ filter : 23, stride : 1

$b=2 \Rightarrow a=6 \quad \Rightarrow$ filter : 6, stride : 2

(e) A mini-CNN is defined below. All the pixels in all Ms have the same value $1/9$. If all the parameters of the CNN have the same value 1, find the pixel value z. (Use the class convention that the biases of the convolutional filters are zero.)

$M_{1-3}(9 \text{ by } 9)$, $C_{1-2}(3 \text{ by } 3)$, stride 2, Y, 2 by 2 Maxpooling, $YS \longrightarrow z$

$Y: (9-3)/2+1=4$

$Y_{i,j} = (1 \times \frac{1}{9}) \times (3 \times 3) \times 3 = 3$

$Y = \begin{bmatrix} 3 & 3 & 3 & 3 \\ 3 & 3 & 3 & 3 \\ 3 & 3 & 3 & 3 \\ 3 & 3 & 3 & 3 \end{bmatrix}$ (2)

$YS = \begin{bmatrix} 3 & 3 \\ 3 & 3 \end{bmatrix}$ (2)

$z = 2 \times (3 \times 4) + 1 = 25.$