CMPE 258 Midterm Exam Version A
2021S (close book, close notes)

First Name: ________________ Last Name: ______________
Student ID: ________________ Email:______________________ (Total 30 points)

Honor Code: All students agree, individually and collectively, that they will not give or receive unpermitted aid during examinations. Engaging in such unpermitted actions is considered cheating. By my signature, I affirm that I have adhered to the spirit of the Honor Code.

Signature:___________________________________Date:_________________

Please Read: instructions for Zoon based online exam.
1. this is an one hour exam, starting from 3:00 and last till 4:00;
2. use cellphone to scan your papers, save them as pdf file, then you must convert it zip file;
3. Submit the zip file on line before 4:15, late submission will make the paper disqualified; After 4:15 no submission will be accepted (based on the time stamp of the submission). Thank you and stay safe**.**

There are total 3 questions in this midterm, and the questions **start next page**.

QUESTION 1 (10 points) Based on the project of MNIST Convolutional Neural Network for handwritten digits recognition, answer the following questions:

1.1 (2 pts) Describe MNIST CNN architecture by using C1 for the 1st convolutional layer, M1 for the 1st max pooling layer, etc., based on the Tensorflow model.summary() output (as shown in the figure below), and explain why the output shape of the convolution is 26x26 while the input image is 28x28?

1.2 (3 pts) Suppose an image layer I(x,y) is given in the following figure and a 3x3 Laplace of Gaussian (LoG) kernel is to be utilized to perform convolution. Find the kernel coefficients LoG(x,y), assuming sigma=1 (Note: the formula for LoG(x,y) is given in the Appendix, and LoG(0,1)=LoG(0,-1), and LoG(0,1)=LoG(1,0), so the computation can be simplified)?

1.3 (4 pts) Compute 2D convolution by using LoG kernel at image plane location (2,2), assuming the origin of the image at top left hand corner is (1,1).

1.4 (1 pt) First Write a line of python code using TF Keras to save a trained MNIST model, then write a line of python code to load the trained MNIST model.

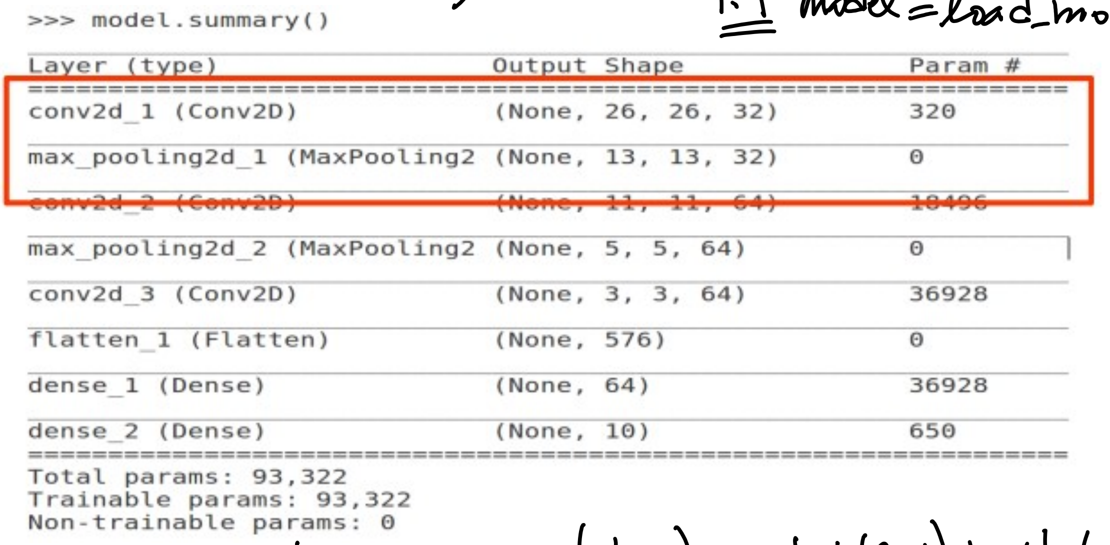
2	0	0	10
0	2	0	10
0	0	0	10
1	1	0	10

Sol. 1.1. Architecture: C1-M1-C2-M2-C3-F-D1-D2

Convolution Kernel size is 3x3, Hence I(x,y) after convolution lost the top & Button Row, as well as left & Right col. So the size 28x28 becomes 26x26

1.2 $LoG(0,0) = \frac{-2}{\sqrt{2\pi}}$, $LoG(1,0) = LoG(0,1) = -\frac{1}{\sqrt{2\pi}}$, $LoG(1,1) = 0$

1.4 model.save('my.h5')
model=load_model('my.h5')



```
>>> model.summary()
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 26, 26, 32)        320
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 13, 13, 32)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 11, 11, 64)        18496
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 5, 5, 64)          0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 3, 3, 64)          36928
_________________________________________________________________
flatten_1 (Flatten)          (None, 576)               0
_________________________________________________________________
dense_1 (Dense)              (None, 64)                36928
_________________________________________________________________
dense_2 (Dense)              (None, 10)                650
=================================================================
Total params: 93,322
Trainable params: 93,322
Non-trainable params: 0
```

1.3 $$LoG(x,y)*I(x,y)\Big|_{x=y=2} = 2*LoG(-1,-1)+0*LoG(0,-1)+0*LoG(1,-1)$$
$$+ 0*LoG(-1,0)+2*LoG(0,0)+0*LoG(1,0)$$
$$+ 0*LoG(-1,1)+0*LoG(0,1)+0*LoG(1,1)$$

$$= 2*LoG(-1,-1)+2*LoG(0,0) = 2*LoG(1,1)+2*\frac{-2}{\sqrt{2\pi}} = -\frac{4}{\sqrt{2\pi}}$$,,

QUESTION 2 (10 points). Gradient descent technique is applied to train Neural Networks, use this technique to answer the following questions:
2.1 (3 pts) Suppose a feed forward NN is given in the figure below with an activation function f( ), for each neuron output, find
h1 = ?
h2 = ?
o1 = ?

$\underline{2.1}$ $h_1 = f(x_1 w_1 + x_2 w_2 + b_1)$, $h_2 = f(x_1 w_3 + x_2 w_4 + b_2)$

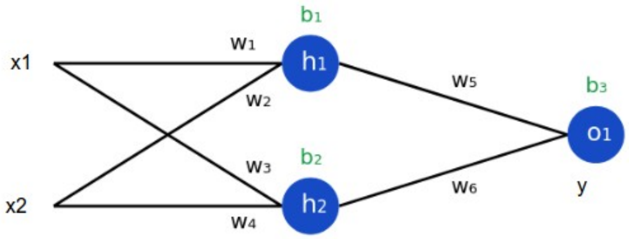
$O_1 = f(h_1 w_5 + h_2 w_6 + b_3)$

2.2 (1 pt) What is the weight w_i update equation based on gradient descent technique? Define a loss function as MSE (Mean Square Error shown below), choose Sigmoid activation function (See Appendix);
2.3 (5 pts) Find the updated w1 by using gradient descent technique, assuming x1=1, x2=-1, and all the w's are at the initial value 0.5.
2.4 (1 pt) Using Taylor expansion up to the first linear term (see Appendix) to explain why gradient descent technique will lead to the reduction of the loss function. (Note you can use a simplified two dimension case, such as x1 and x2, for your analysis and explanation).

$\underline{2.2}$ $W_i^+ = W_i^- - \eta \frac{\partial L}{\partial W_i}$,

Lossfunction (MSE)

$$L \triangleq \sum_{i=1}^{N} (y_{true,i} - y_{pred,i})^2$$



$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

$\underline{2.3}$ First, $\frac{\partial L}{\partial w_1} = \frac{\partial L}{\partial y_{pred}} \cdot \frac{\partial y_{pred}}{\partial h_1} \cdot \frac{\partial h_1}{\partial w_1}$ $= \frac{\partial}{\partial w_1}\left[\sum_{i=1}^{N}(y_{true,i} - y_{pred,i})^2\right]$

$-2 \cdot \sum_{i=1}^{N}(y_{true,i} - y_{pred,i}) \frac{\partial y_{pred,i}}{\partial h_1} \cdot \frac{\partial h_1}{\partial w_1}$, where $\frac{\partial y_{pred,i}}{\partial h_1} = \frac{\partial}{h_1} f(h_1 w_5 + h_2 w_6 + b_3) = f'_{h_1} \cdot w_5$

and $\frac{\partial h_1}{\partial w_1} = \frac{\partial}{\partial w_1} f(x_1 w_1 + x_2 w_2 + b_1) = f'_{x_1} x_1$, Activation function f and its derivative f' are given in the Appendix (As a Sigmoid function), hence,

$\frac{\partial L}{\partial w_1} = -2 \sum_{i=1}^{N}(y_{x,i} - y_{pred,i}) f'_{h_1} W5 \cdot f'_{x_1} X_1$, where $f'_{x_1} = \frac{e^{-1}}{(1+e^{-1})^2}$, and $f'_{h_1} = f'_{x_1}$

Therefore, $\frac{\partial L}{\partial w_1} = -2\left(\sum_{i=1}^{N} 1 - 0.5\right)\left[\frac{e^{-1}}{(1+e^{-1})^2}\right]^2 0.5^2 = -2N * 0.5\left[\frac{e^{-1}}{(1+e^{-1})^2}\right]^2 0.5^2$

$= -0.25\left[\frac{e^{-1}}{(1+e^{-1})^2}\right]^2$ (Let N=1 for 1 update). $\underline{2.4}$. See Appendix

QUESTION 3 (10 points). Based on the design for handwritten digits recognition, answer the following questions:

3.1 (2 pts) Design an implementation technique by providing step by step detailed flow chart for handwritten digits recognition?

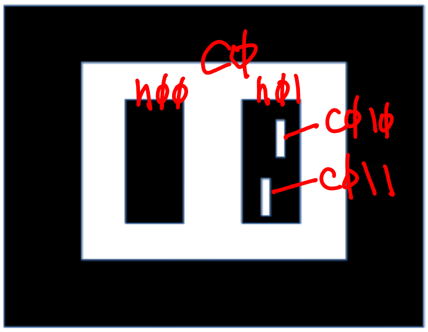
3.2 (2 pts) Find a contour (2 pts) of the image belwo by using CV_RETR_TREE algorithm.

Sol. 3.2, Cϕ

hϕϕ – hϕ1

cϕ1ϕ – cϕ11

(Note, the Labels are marked on the Binary Image Pattern)



3.3 (3 pts) Find x-bar of the following binary image during the localization of ROI operation;

3.4 (3 pts) In order to find orientation of the object given in the following binary image, compute a (see Appendix).

1			
	1		

$$\bar{X} = \frac{\sum_{K_2=1}^{4}\sum_{K_1=1}^{4} K_1 B(K_1,K_2)}{\sum_{K_2=1}^{4}\sum_{K_1=1}^{4} B(K_1,K_2)} \quad \ldots (1)$$

where

$\sum_{K_2=1}^{4}\sum_{K_1=1}^{4} B(K_1,K_2) = 2$, and $\sum_{K_2=1}^{4}\sum_{K_1=1}^{4} K_1 B(K_1,K_2) = 1\cdot B(1,1) + 2\cdot B(2,2)$

$= 1\cdot 1 + 2\cdot 1 = 1 + 2 = 3$

(Note: Rest of $B(K_1,K_2) = 0$)

$\therefore \bar{X} = 3/2 = 1,5$

Now Compute a

$$a = \frac{\sum_{K_2=1}^{4}\sum_{K_1=1}^{4} (K_1-\bar{x})^2 B(K_1,K_2)}{\sum_{K_2=1}^{4}\sum_{K_1=1}^{4} B(K_1,K_2)}$$

where $\sum_{K_2=1}^{4}\sum_{K_1=1}^{4} (K_1-\bar{x})^2 B(K_1,K_2)$

$= (1-\bar{x})^2 \cdot B(1,1) + (2-\bar{x})^2 B(2,2)$

$= 0,5^2 \cdot 1 + 0,5^2 \cdot 1 = 2\times 0.25 = 0,5$

Hence, $a = 0,5/1,5 = 1/3$ //

APPENDIX A. CNN Architecture

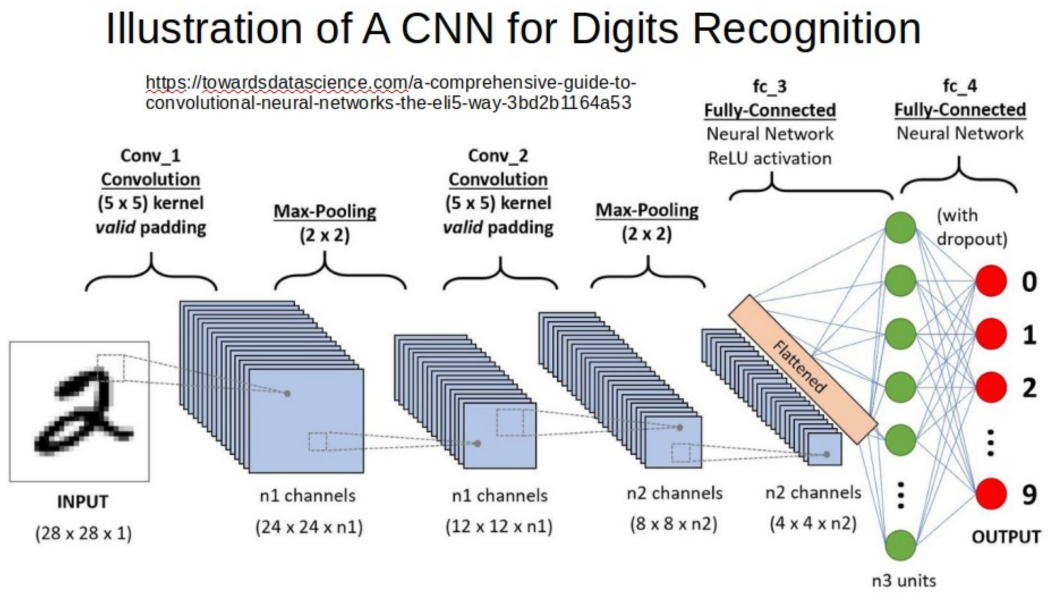


Illustration of A CNN for Digits Recognition

https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

Harry Li, Ph.D. 2018

Orientation

$$\tan 2\phi \triangleq \frac{b}{a-c}$$

Moments:

$$M_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x^p y^q f(x,y)\,dx\,dy$$

$$a = \iint_\Omega (x-\bar{x})^2 B(x,y)\,dxdy \quad \cdots (2)$$

Central Moments:

$$\mu_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} (x-\bar{x})^p (y-\bar{y})^q f(x,y)\,dx\,dy$$

$$b = \iint_\Omega 2(x-\bar{x})(y-\bar{y}) B(x,y)\,dxdy \quad \cdots (3)$$

$$c = \iint_\Omega (y-\bar{y})^2 B(x,y)\,dxdy \quad \cdots (4)$$

For 1,2: $LoG(0,0) = -\frac{2\sigma^2}{\sqrt{2\pi}\sigma^5} = -\frac{2}{\sqrt{2\pi}}$

APPENDIX B. LoG(x,y) kernel

$LoG(0,1) = -\frac{1}{\sqrt{2\pi}} = LoG(1,0)$

$LoG(1,1) = 0$

$$\nabla^2 G(x,y) = \frac{x^2+y^2-2\sigma^2}{\sqrt{2\pi}\,\sigma^3} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

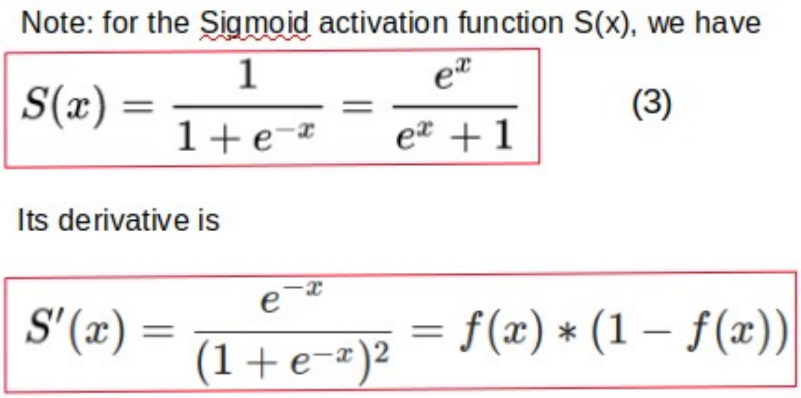
For 1,4: (from keras import models)

model.save('my.h5')

(from keras.models import load_model)

model=load_model('my.h5')

APPENDIX C. Sigmoid Activation Function and Example for Gradient Descent Based Training Formula



Note: for the Sigmoid activation function S(x), we have

$$\boxed{S(x) = \frac{1}{1+e^{-x}} = \frac{e^x}{e^x+1}} \qquad (3)$$

Its derivative is

$$\boxed{S'(x) = \frac{e^{-x}}{(1+e^{-x})^2} = f(x) * (1 - f(x))}$$

For 2.3: $h_1 = f(x_1 w_1 + x_2 w_2 + b_1) = f(0{,}5*0{,}5+0{,}5*0{,}5+0{,}5) = f(1)$ so, $f'_{x_1} = f'(1) = f(1)(1-f(1)) = \frac{e^{-1}}{(1+e^{-1})^2}$; Similarly, $o_1 = f(h_1 w_5 + h_2 w_6 + b_3) = f(1)$, so $f'_{h_1} = f'_{x_1}$

$$w_1 \leftarrow w_1 - \eta \frac{\partial L}{\partial w_1} \qquad (4)$$

APPENDIX D.

$$f(x_1, x_2) \simeq f(a,b) + \frac{\partial f}{\partial x_1}(x_1 - a) + \frac{\partial f}{\partial x_2}(x_2 - b) \quad \cdots (1)$$

For 2.4 Let $f(x_1,x_2)$ be a Loss function, $f(x_1,x_2)$ is a new One-Step updated function Comparing to $f(a,b)$, so, for $f(x_1,x_2) - f(a,b) < 0$, the updated independent variable $x_1 - a = -\frac{\partial f}{\partial x_1}$, $x_2 - b = -\frac{\partial f}{\partial x_2}$, which leads to $-\left(\frac{\partial f}{\partial x_1}\right)^2 - \left(\frac{\partial f}{\partial x_2}\right)^2 < 0$

(END) Hence, $f(x_1,x_2) < f(a,b)$. //