Deep Learning HW #2 Instructor: Shusen Pu

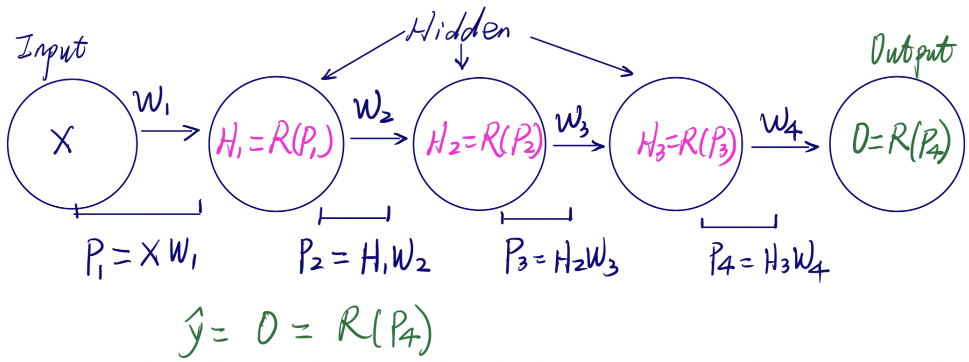
Student Name (print): Farooq Mahmud **Group #** 10

Question:	1	2	3	Total
Points:	10	10	20	40
Score:				

Instructions: This homework assignment is divided into three sections:

- The first section requires you to calculate derivatives either by typing or handwriting
- The second section, labeled as problem 2, combines traditional problem-solving with programming. You will begin by manually working out the problem, then you'll develop your own Python code to further solve it. Please submit an *ipynb* file that successfully runs and solves parts c through f, including the generation of relevant plots. This submission should also include a master script that triggers all results and additional functions that support the script.
- The third section repeats the format of last week's coding assignment.

1. (10 points) Let the cost function $C = \frac{1}{2}(\hat{y} - y)^2$ for the following neural network model with input x and output $\hat{y} = O$ (in green).



(a) What is the derivative of the cost function with respect to W_4?

$$(\hat{y} - y)R_4'(P_4)H_3$$

(b) What is the derivative of the cost function with respect to W_3?

$$(\hat{y} - y)R_4'(P_4)w_4R_3'(P_3)H_2$$

(c) What is the derivative of the cost function with respect to W_1?

$$(\hat{y} - y)R_4'(P_4)w_4R_3'(P_3)w_3R_2'(P_2)w_2R_1'(P_1)x$$

(d) What is the output layer error for this network model?

$$\hat{y} - y$$

(e) What is the derivative of the cost function with respect to the hidden layer input P_2?

$$(\hat{y} - y)R_4'(P_4)w_4R_3'(P_3)w_3R_2'(P_2)$$

2. (10 points) **Newton's Method** Use Newton's method to approximate $\sqrt[4]{30}$ and stop the iteration when it is accurate up to 7 decimals.

   (a) Form this question into a mathematical problem

   $$x^4 - 30 = 0$$

   (b) Use Newton's Method to find the (iterative) updating formula for x_n based on x_{n-1}

   $$x_n = x_{n-1} - \left(\frac{x_{n-1}^4 - 30}{4x_{n-1}^3}\right)$$

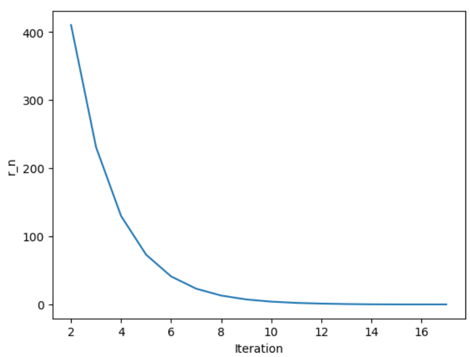
   (c) Write a Python function that finds the solution to this problem with a tolerance $\tau = 10^{-8}$, where $\tau = \mid x_n - x_{n-1} \mid$. Instructions:

   - Write your code from scratch without using any packages
   - Your code should be flexible so that you can change the initial conditions and tolerance

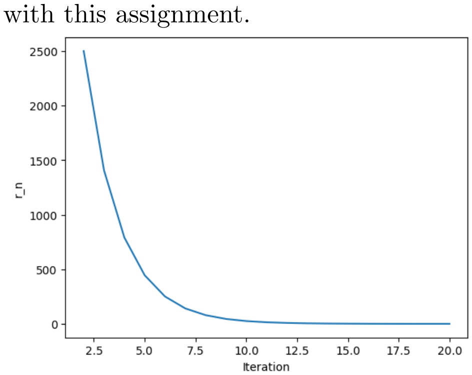
   (d) Set your initial condition $x_0 = 81$. How many iterations are needed? Write down all the intermediate results (i.e., output $x_1, x_2, \cdots, x_n$).

   17 iterations are needed

   (e) Let $r_n = (x_n - x_{n-1})^2$, plot r_n in Python and save it in eps (Encapsulated PostScript) format (attach the plot with this homework).



   (f) Change the initial condition to $x_0 = 200$, $\tau = 10^{-9}$ and repeat steps (d)-(e). Also, attach the plot with this assignment.



   20 iterations needed

   (g) Upload your code for this problem on Canvas under this assignment. I will subsequently review and run your code in order to grade your assignment.

3. (20 points) Complete the coding tasks in the file titled *Homework2.ipynb*
   Instructions:

   - The code with other example codes are available in Google Drive:
     `https://drive.google.com/drive/folders/1Gc6XcHc7Qr3EGQ_dzpkEdeNdwHdSkbr0?usp=sharing`
   - Once you've finished, save your work and rename the file by appending your initials. For instance, if your name is Shusen Pu, you should rename the file as "Homework2-SP.ipynb".
   - Make sure all your inputs in the code are saved. Then close and reopen the renamed file (in my case, "Homework2-SP.ipynb") to confirm that all your answers have been saved properly.
   - Upload your code on Canvas under this assignment.

   I will subsequently review and run your code in order to grade your assignment.

Given:
$\hat{y} = O = R_4(P_4)$
$P_4 = H_3 w_4$
$H_3 = R_3(P_3)$
$P_3 = H_2 w_3$
$H_2 = R_2(P_2)$
$P_2 = H_1 w_2$
$H_1 = R_1(P_1)$
$P_1 = x w_1$
$C = \frac{1}{2}(\hat{y} - y)^2$

1 Find $\frac{d}{dw_4}[C]$:

$=(\hat{y} - y)\frac{d}{dw_4}\hat{y}$
$= (\hat{y} - y)\frac{d}{dw_4}[R_4(P_4)]$
$= (\hat{y} - y)R_4'(P_4)\frac{d}{dw_4}[H_3 w_4]$
$= (\hat{y} - y)R_4'(P_4)H_3$

2 Find $\frac{d}{dw_3}[C]$:

$=(\hat{y} - y)R_4'(P_4)\frac{d}{dw_3}[H_3 w_4]$
$= (\hat{y} - y)R_4'(P_4)w_4\frac{d}{dw_3}[H_3]$
$= (\hat{y} - y)R_4'(P_4)w_4\frac{d}{dw_3}[R_3(P_3)] = (\hat{y} - y)R_4'(P_4)w_4 R_3'(P_3)\frac{d}{dw_3}[P_3]$
$= (\hat{y} - y)R_4'(P_4)w_4 R_3'(P_3)\frac{d}{dw_3}[H_2 w_3]$
$= (\hat{y} - y)R_4'(P_4)w_4 R_3'(P_3)H_2$

3 Find $\frac{d}{dw_1}[C]$:

$=(\hat{y} - y)R_4'(P_4)w_4 R_3'(P_3)\frac{d}{dw_3}[H_2 w_3]$
$= (\hat{y} - y)R_4'(P_4)w_4 R_3'(P_3)w_3\frac{d}{dw_1}[H_2]$
$= (\hat{y} - y)R_4'(P_4)w_4 R_3'(P_3)w_3\frac{d}{dw_1}[R_2(P_2)]$
$= (\hat{y} - y)R_4'(P_4)w_4 R_3'(P_3)w_3 R_2'(P_2)\frac{d}{dw_1}[P_2]$
$= (\hat{y} - y)R_4'(P_4)w_4 R_3'(P_3)w_3 R_2'(P_2)\frac{d}{dw_1}[H_1 w_2]$
$= (\hat{y} - y)R_4'(P_4)w_4 R_3'(P_3)w_3 R_2'(P_2)w_2\frac{d}{dw_1}[R_1(P_1)]$
$= (\hat{y} - y)R_4'(P_4)w_4 R_3'(P_3)w_3 R_2'(P_2)w_2 R_1'(P_1)\frac{d}{dw_1}[P_1]$
$= (\hat{y} - y)R_4'(P_4)w_4 R_3'(P_3)w_3 R_2'(P_2)w_2 R_1'(P_1)\frac{d}{dw_1}[x w_1]$
$= (\hat{y} - y)R_4'(P_4)w_4 R_3'(P_3)w_3 R_2'(P_2)w_2 R_1'(P_1)x$

4 Output layer error

$\hat{y} - y$

5 Find $\frac{d}{dP_2}[C]$:

$=(\hat{y} - y)R_4'(P_4)w_4R_3'(P_3)w_3\frac{d}{dP_2}[R_2(P_2)]$
$= (\hat{y} - y)R_4'(P_4)w_4R_3'(P_3)w_3R_2'(P_2)$