DELTA Page No. Date

Assignment - 1

Sol^n: 1 (a) Taylor series expansion for a function $f(x)$ at $x = 0$ is given by:

$$f(x) = \sum_{n=0}^{\infty} \frac{f^n(0)}{n!} \cdot x^n \longrightarrow (1)$$

$$f(x) = e^x \quad \text{(given)}$$

Let's make a table for the function till 3rd order derivative:

$$\begin{aligned}
f(x) &= e^x & \Rightarrow f(0) &= e^0 = 1\\
f'(x) &= e^x & \Rightarrow f'(0) &= e^0 = 1\\
f''(x) &= e^x & \Rightarrow f''(0) &= e^0 = 1\\
f'''(x) &= e^x & \Rightarrow f'''(0) &= e^0 = 1\\
\cdots & \quad \cdots & & \\
f^n(0) &= e^x & \Rightarrow f^n(0) &= e^0 = 1.
\end{aligned}$$

For a n^{th} degree taylor polynomial $f^n(0) = 1$.
Using this value in (1),

$$\Rightarrow \frac{1 \cdot x^0}{0!} + \frac{1 \cdot x}{1!} + \frac{1 \cdot x^2}{2!} + \cdots + \frac{1 \cdot x^n}{n!}$$

$$\left[T_n(x) = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots + \frac{x^n}{n!} \right]$$

DELTA Page No.
Date / /

Sol? 1(b) As we know the taylor series expansion:

$$f(x) = e^x$$
$$T_n(x) = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots + \frac{x^n}{n!}$$

- To plot a graph to see what happens as we progressively take higher ~~terms~~ order terms:

↳ create a sample space of 150 points from -4 to 4. The various graphs will be plot using these points.

↳ Take an output dummy array with a length of 5 (max order term is 4), the graphs will be plot on the points added to this array.

↳ Loop on a range of $0 \to 5$, for index of 0, all the values would be 1, as the first position value in taylor series is 1, for the rest, add the previous index value plus, $(x^i / i!)$. Store all the values in the dummy output array.

↳ Use the dummy array to plot the graphs. and 1 actual e^x graph that we are trying model using the Taylor series

↳ Graph : img-1(b).png

Graph code: 1-b.py / 1-b.ipynb

Sol: 2(b) $\nabla w = \begin{bmatrix} \frac{\partial J}{\partial w_1} \\ \frac{\partial J}{\partial w_2} \end{bmatrix}$

$$J = \frac{1}{2}\left[(w_2 - w_1)^2 + (1 - w_1)^2\right]$$

Finding the partial derivatives,

- $$\frac{\partial J}{\partial w_1} = \frac{\partial}{\partial w_1}\frac{1}{2}\left[(w_2 - w_1)^2 + (1 - w_1)^2\right]$$

$$= \frac{1}{2}\left[2(w_2 - w_1)\frac{\partial}{\partial w_1}(w_2 - w_1) + 2(1 - w_1)\frac{\partial}{\partial w_1}(1 - w_1)\right]$$

$$= \frac{1}{\cancel{2}}\left[\cancel{2}(w_2 - w_1)(-1) + \cancel{2}(1 - w_1)(-1)\right]$$

$$= (w_1 - w_2) + (w_1 - 1)$$

$$\frac{\partial J}{\partial w_1} = 2w_1 - w_2 - 1 \longrightarrow ①$$

- $$\frac{\partial J}{\partial w_2} = \frac{\partial}{\partial w_2}\cdot\frac{1}{2}\left[(w_2 - w_1)^2 + (1 - w_1)^2\right]$$

$$= \frac{1}{2}\left[2(w_2 - w_1)\frac{\partial}{\partial w_2}(w_2 - w_1) + 2(1 - w_1)\cdot\frac{\partial}{\partial w_2}(1 - w_1)\right]$$

P.T.O

$$= \frac{1}{\cancel{2}}\left[\cancel{2}(w_2 - w_1)(1) + \cancel{2}(1 - w_1)(0 - 0)\right]$$

$$\frac{\partial J}{\partial w_2} = w_2 - w_1 \longrightarrow (2).$$

from (1) and (2),

$$\nabla w = \begin{bmatrix} 2w_1 - w_2 - 1 \\ w_2 - w_1 \end{bmatrix}$$

DELTA Page No.
Date

Sol: 2(d) From part (B), we got the analytical value of the gradient vector,

$$\nabla w = \begin{bmatrix} 2w_1 - w_2 - 1 \\ w_2 - w_1 \end{bmatrix} = \begin{bmatrix} \frac{\partial J}{\partial w_1} \\ \frac{\partial J}{\partial w_2} \end{bmatrix}$$

To minimize the error function, the gradient vector should be equal to 0, and the 2 equations will equal to 0 and thus we can find the weight vectors.

$$\nabla w = 0 \Rightarrow \begin{bmatrix} 2w_1 - w_2 - 1 \\ w_2 - w_1 \end{bmatrix} = 0.$$

- $2w_1 - w_2 - 1 = 0 \rightarrow (1)$
- $w_2 - w_1 = 0 \rightarrow (2)$

Adding (1) and (2).

$$\begin{array}{lll} 2w_1 & - \cancel{w_2} & = 1 \\ -w_1 & + \cancel{w_2} & = 0 \\ + & + & + \\ \hline w_1 & & = 1. \end{array} \rightarrow (3).$$

Using the value of (3) in (2),

$$w_2 - w_1 = 0$$

$$\Rightarrow w_2 - 1 = 0 \quad \Rightarrow \quad \underline{w_2 = 0} \quad \rightarrow (4)$$

Therefore, from ③ and ④,

$w_1 = 1$
$w_2 = 1$

So, the loss function will be minimum at $(w_1, w_2) = (1, 1)$

Sol^n: 2 (a) Contour plot : contour-2(a).png

Contour plot code: ~~[illegible]~~ 2-a.py / 2-a.ipynb

↳ created an initial range of 150 points from -50 to 50.

↳ Using those 150 points, created a mesh grid [X, Y].

↳ created a cost function with the X, Y values

↳ plot the contour plot using matplotlib contour function.

Sol: 2(c) We have to create at arbitary locations:

For this, we can use some random arbitary values for the weight vectors.

$$\nabla W = \begin{bmatrix} \frac{\partial J}{\partial w_1} \\ \frac{\partial J}{\partial w_2} \end{bmatrix} = \begin{bmatrix} 2w_1 - w_2 - 1 \\ w_2 - w_1 \end{bmatrix}$$

We got the upper value from part (B).

- Using $w_1 = 2$ and $w_2 = 5$

$$\nabla W = \begin{bmatrix} 2(2) - 5 - 1 \\ 5 - 2 \end{bmatrix} = \begin{bmatrix} 4 - 6 \\ 3 \end{bmatrix} = \begin{bmatrix} -2 \\ 3 \end{bmatrix} \rightarrow (1)$$

- Using $w_1 = 10$, $w_2 = 5$

$$\nabla W = \begin{bmatrix} 2(10) - 5 - 1 \\ 5 - 10 \end{bmatrix} = \begin{bmatrix} 20 - 6 \\ -5 \end{bmatrix} = \begin{bmatrix} 14 \\ -5 \end{bmatrix} \rightarrow (2)$$

- Using $w_1 = 0$, $w_2 = 5$

$$\nabla W = \begin{bmatrix} 2(0) - 5 - 1 \\ 5 - 0 \end{bmatrix} = \begin{bmatrix} 0 - 6 \\ 5 \end{bmatrix} = \begin{bmatrix} -6 \\ 5 \end{bmatrix} \rightarrow (3)$$

From ①,

$w_1 = 2$, $w_2 = 5$, $\frac{\partial J}{\partial w_1} = -2$, $\frac{\partial J}{\partial w_2} = 3$

From ②,

$w_1 = 10$, $w_2 = 5$, $\frac{\partial J}{\partial w_1} = 14$, $\frac{\partial J}{\partial w_2} = -5$

From ③,

$w_1 = 0$, $w_2 = 5$, $\frac{\partial J}{\partial w_1} = -6$, $\frac{\partial J}{\partial w_2} = 5$

↳ we will first place a contour map, similar to what we did in part (A).

↳ After placing the contour plot, we will add 3 arrows (for directions) at the points we received from ①, ② and ③.

→ for code : 2-c.py | 2-c.ipynb

→ for plot : contour-2(c).png.