1.1 Introduction:

The aim of this project is to implement the artificial neural network algorithm to solve the multiclass (10 class) classification problem for a given dataset.

This assignment teaches us how the algorithms written in the sklearn library actually works and in depth mathematical intuition behind these algorithms.

The dataset used in this project consists of six independent features i.e. X1, X2 ,X3 ,X4 ,X5 ,X6 and a dependent feature Y. Our goal is to use these independent features to predict the values of dependent feature

1.2 Code Design, Methodology and Derivations:

The code can be divided into following subsections for clear understanding:

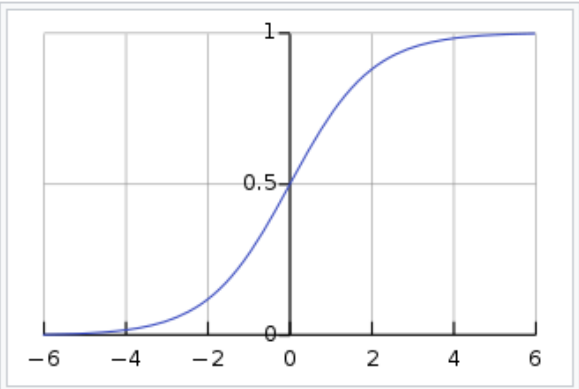
A. Importing Required Libraries:
You need to install and import numpy, pandas and matplotlib libraries to successfully run this project

B. Data Preprocessing:

1. First we are reading the csv file using the pandas library
2. Then we are normalizing the data set using
   X -> (X - mean)/(standard deviation)
3. Shuffling the data
4. Performing test train split so that we use 70% of the data for training and 30% of the data for testing.

C. Data Learning With Two Layer Artificial Neural Network
Firstly we have defined the sigmoid function.Mathematical formula for sigmoid function is

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



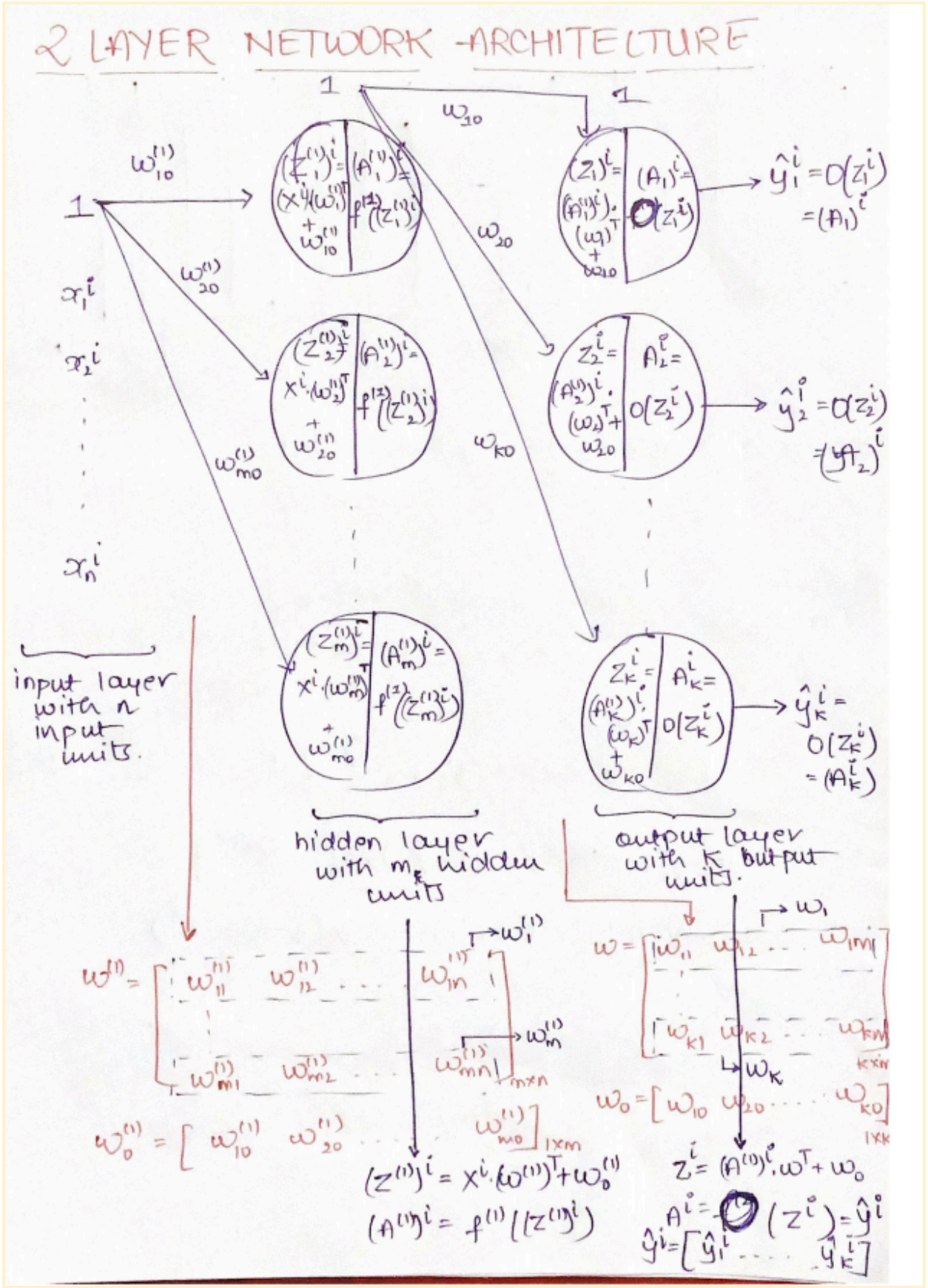
The logistic curve

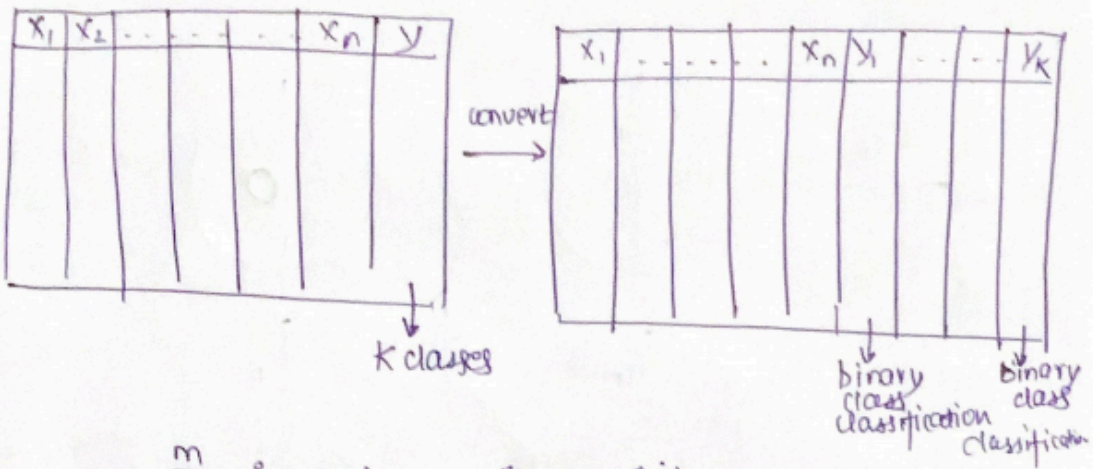
Next we defined the derivative of the sigmoid function. Mathematical formula for first derivative of sigmoid function is given by

$$S'(x) = S(x) *(1 - S(x))$$

Now a function called stochastic gradient descent is defined which is the core of the entire algorithm.It takes the data, learning rate, no of iterations and no of hidden units as inputs.In this we have used stochastic gradient descent algorithm to optimize the unconstrained optimization problem formed in ANN.Since we are using stochastic gradient descent algorithm the cost function due to just one example.

This function can be further divided into following parts , namely: forward propagation , backward propagation and weight updation.The formulas used in all these can be understood by understanding the complete math intuition of the algorithm provided in notes below and the coding is done using similar notations for easy understanding.


2 LAYER NETWORK -ARCHITECTURE
input layer with n input units.
hidden layer with m hidden units
output layer with K output units.
$(Z_1^{(1)})^i = x^i \cdot (w_1^{(1)})^T + w_{10}^{(1)}$
$(A_1^{(1)})^i = f^{(1)}((Z_1^{(1)})^i)$
$(Z_1)^i = (A_1^{(1)})^i \cdot (w_1)^T + w_{10}$
$(A_1)^i = O(Z_1^i)$
$\hat{y}_1^i = O(Z_1^i) = (A_1)^i$
$(Z_2^{(1)})^i = x^i \cdot (w_2^{(1)})^T + w_{20}^{(1)}$
$(A_2^{(1)})^i = f^{(1)}((Z_2^{(1)})^i)$
$Z_2^i = (A_2^{(1)})^i \cdot (w_2)^T + w_{20}$
$A_2^i = O(Z_2^i)$
$\hat{y}_2^i = O(Z_2^i) = (A_2)^i$
$(Z_m^{(1)})^i = x^i \cdot (w_m^{(1)})^T + w_{m0}^{(1)}$
$(A_m^{(1)})^i = f^{(1)}((Z_m^{(1)})^i)$
$Z_K^i = (A_K^{(1)})^i \cdot (w_K)^T + w_{K0}$
$A_K^i = O(Z_K^i)$
$\hat{y}_K^i = O(Z_K^i) = (A_K^i)$
$w^{(1)} = \begin{bmatrix} w_{11}^{(1)} & w_{12}^{(1)} & \cdots & w_{1n}^{(1)} \\ \vdots & & & \\ w_{m1}^{(1)} & w_{m2}^{(1)} & \cdots & w_{mn}^{(1)} \end{bmatrix}_{m \times n}$
$w_0^{(1)} = \begin{bmatrix} w_{10}^{(1)} & w_{20}^{(1)} & \cdots & w_{m0}^{(1)} \end{bmatrix}_{1 \times m}$
$(Z^{(1)})^i = x^i \cdot (w^{(1)})^T + w_0^{(1)}$
$(A^{(1)})^i = f^{(1)}((Z^{(1)})^i)$
$w = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1m} \\ \vdots & & & \\ w_{K1} & w_{K2} & \cdots & w_{Km} \end{bmatrix}_{K \times m}$
$w_0 = \begin{bmatrix} w_{10} & w_{20} & \cdots & w_{K0} \end{bmatrix}_{1 \times K}$
$Z^i = (A^{(1)})^i \cdot w^T + w_0$
$A^i = O(Z^i) = \hat{y}^i$
$\hat{y}^i = \begin{bmatrix} \hat{y}_1^i & \cdots & \hat{y}_K^i \end{bmatrix}$



$$\text{Error} = \sum_{i=1}^{m} -y_1^i \ln(\hat{y}_1^i) - (1-y_1^i)\ln(1-\hat{y}_1^i) + \cdots\cdots + \sum_{i=1}^{m} -y_K^i \ln(\hat{y}_K^i) - (1-y_K^i)\ln(1-\hat{y}_K^i)$$

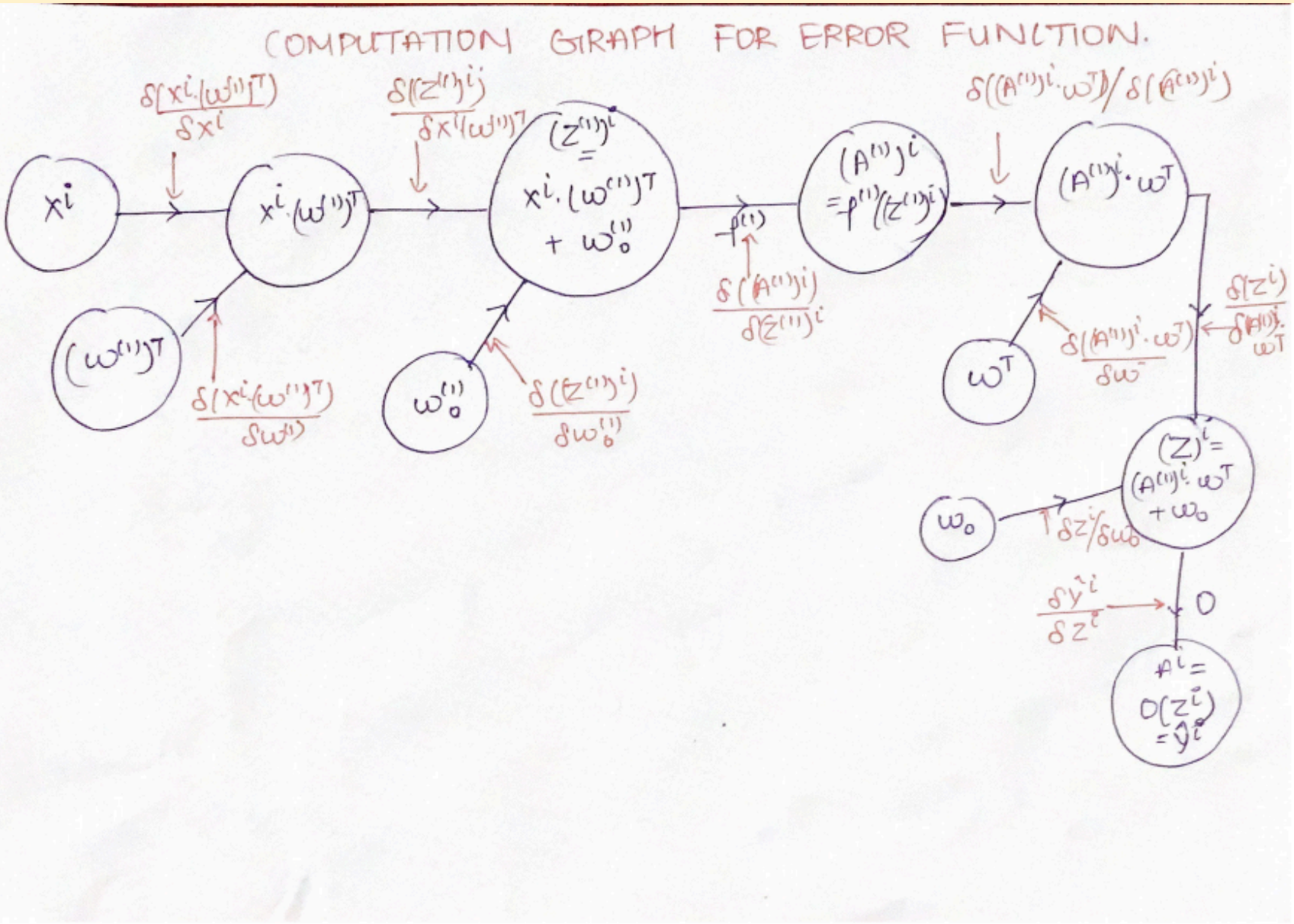
$$\text{Stochastic Error} = -y_1^i \ln(\hat{y}_1^i) - (1-y_1^i)\ln(1-\hat{y}_1^i) + \cdots\cdots + -y_K^i \ln(\hat{y}_K^i) - (1-y_K^i)\ln(1-\hat{y}_K^i)$$

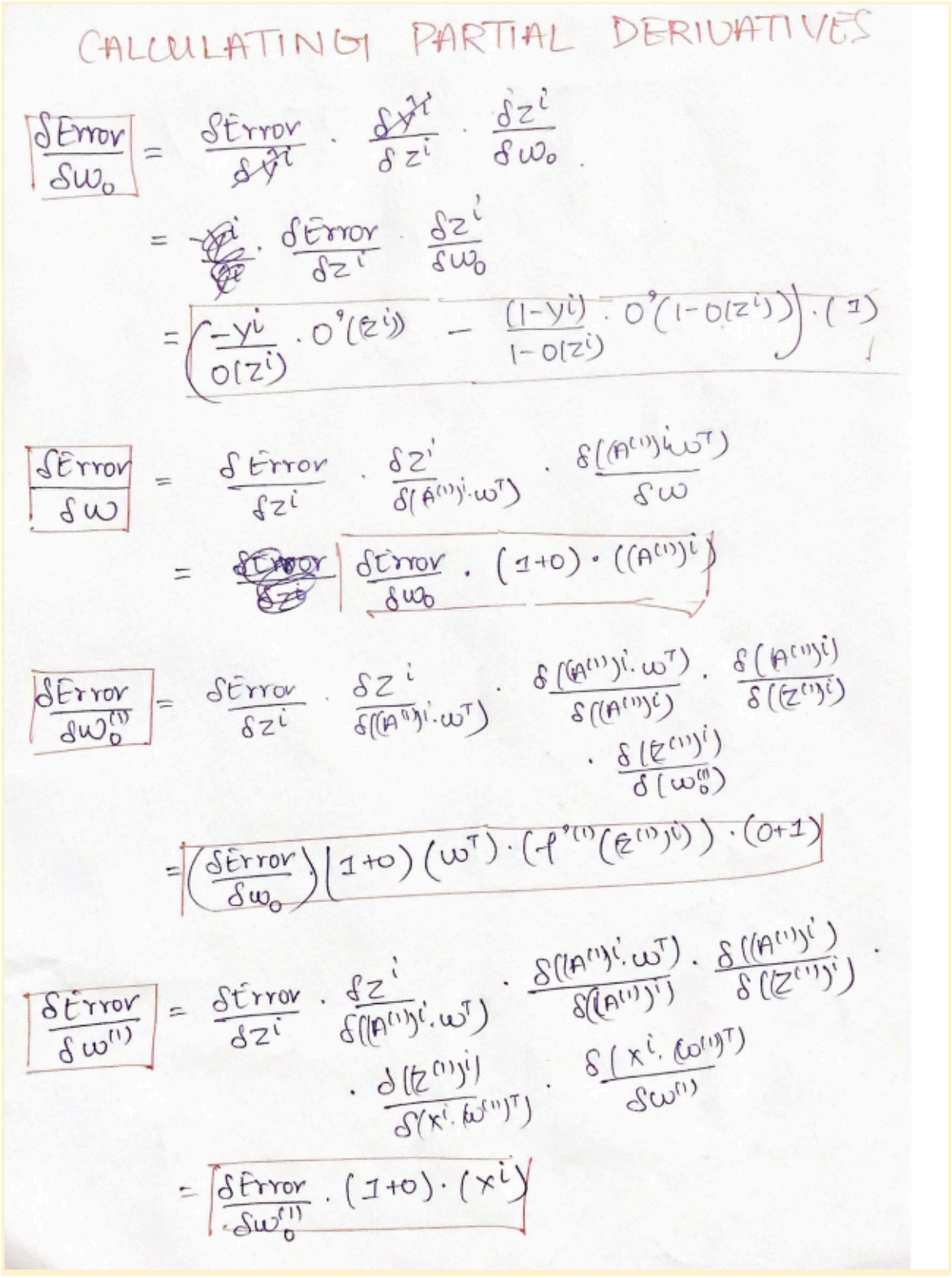
using matrix form

$$y_*^i = [y_1^i \; y_2^i \; \cdots \; y_K^i], \quad \hat{y}^i = [\hat{y}_1^i \; \hat{y}_2^i \; \cdots \; \hat{y}_K^i]$$

$$= -y^i \ln((\hat{y}^i)^T) - (1-y^i)\ln(1-(\hat{y}^i)^T)$$

$$= -y^i \ln(\sigma(z^i)^T) - (1-y^i)\ln(1-\sigma(z^i)^T)$$

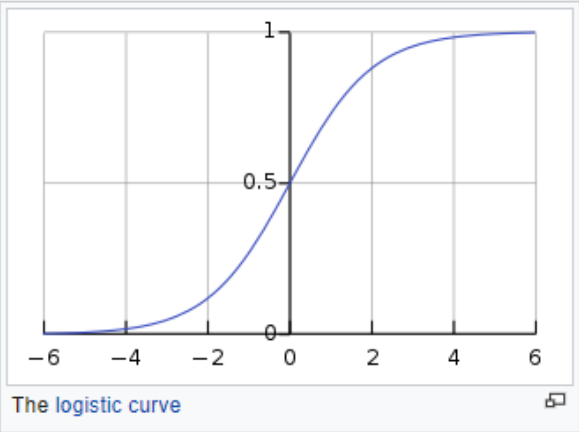

COMPUTATION GRAPH FOR ERROR FUNCTION.
x^i
$x^i.(w^{(1)})^T$
$(w^{(1)})^T$
$w_0^{(1)}$
$(z^{(1)})^i = x^i.(w^{(1)})^T + w_0^{(1)}$
$(A^{(1)})^i = f^{(1)}((z^{(1)})^i)$
$(A^{(1)})^i.w^T$
w^T
$(z)^i = (A^{(1)})^i.w^T + w_0$
w_0
$\hat{y}^i = O(z^i) = \hat{y}^i$
$\delta[x^i.(w^{(1)})^T]/\delta x^i$
$\delta[(z^{(1)})^i]/\delta x^i(w^{(1)})^T$
$\delta((A^{(1)})^i.w^T)/\delta((A^{(1)})^i)$
$\delta(x^i.(w^{(1)})^T)/\delta w^{(1)}$
$\delta((z^{(1)})^i)/\delta w_0^{(1)}$
$f^{(1)}$
$\delta((A^{(1)})^i)/\delta(z^{(1)})^i$
$\delta((A^{(1)})^i.w^T)/\delta w^T$
$\delta[z^i]/\delta((A^{(1)})^i.w^T)$
$\delta z^i/\delta w_0$
$\delta \hat{y}^i/\delta z^i \rightarrow 0$



D. Data Learning Using Three Layer Artificial Neural Network

Firstly we have defined the sigmoid function.Mathematical formula for sigmoid function is

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



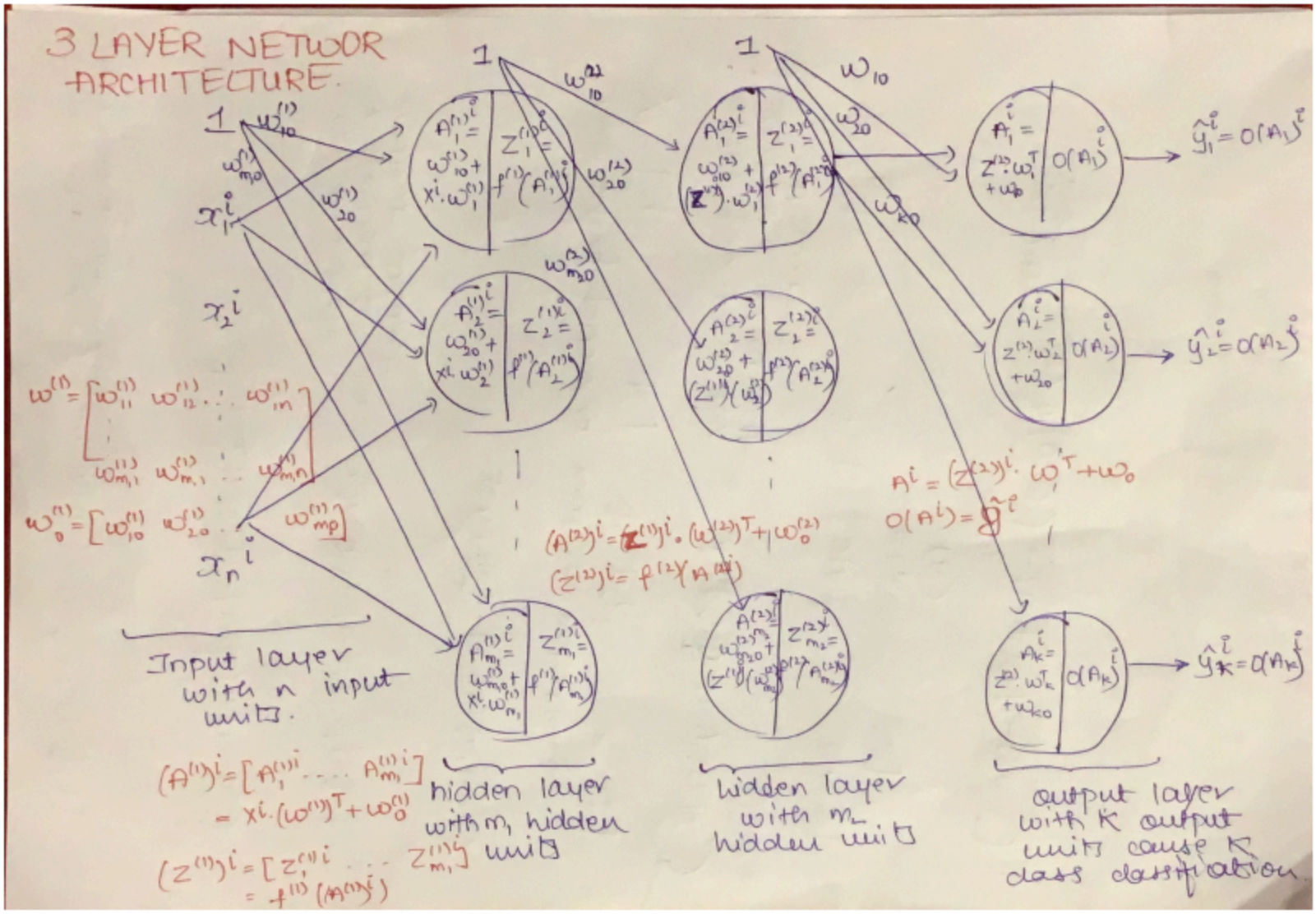
The logistic curve

Next we defined the derivative of the sigmoid function. Mathematical formula for first derivative of sigmoid function is given by

S'(x) = S(x) *(1 - S(x))

Now a function called stochastic gradient descent is defined which is the core of the entire algorithm.It takes the data, learning rate, no of iterations and no of hidden units in hidden layer 1 and no of hidden units in input layer 2 as inputs.In this we have used stochastic gradient descent algorithm to optimize the unconstrained optimization problem formed in ANN.Since we are using stochastic gradient descent algorithm the cost function due to just one example.

This function can be further divided into following parts , namely: forward propagation , backward propagation and weight updation.The formulas used in all these can be understood by understanding the complete math intuition of the algorithm provided in notes below and the coding is done using similar notations for easy understanding.


3 LAYER NETWOOR ARCHITECTURE.
Input layer with n input units.
hidden layer with m1 hidden units
hidden layer with m2 hidden units
output layer with K output units cause ↖ class classification

∴ error function for gradient descent $= \sum_{i=1}^{m} \left(-y_1^i \ln(\hat{y}_1^i) - (1-y_1^i)\ln(1-\hat{y}_1^i)\right)$

$+ \vdots +$

$\sum_{i=1}^{m} \left(-y_K^i \ln(\hat{y}_K^i) - (1-y_K^i)\ln(1-\hat{y}_K^i)\right)$

∴ error function for stochastic gradient descent $= -y_1^i \ln(\hat{y}_1^i) - (1-y_1^i)\ln(1-\hat{y}_1^i)$

$+ \vdots +$

$-y_K^i \ln(\hat{y}_K^i) - (1-y_K^i)\ln(1-\hat{y}_K^i)$

now using following matrix notation

$y^i = [y_1^i \ldots y_K^i]$, $\hat{y}^i = [\hat{y}_1^i \ldots \hat{y}_K^i]$

∴ error function for stochastic gradient descent $= -y^i \ln(\hat{y}^i)^T - (1-y^i)\ln(1-\hat{y}^i)^T$

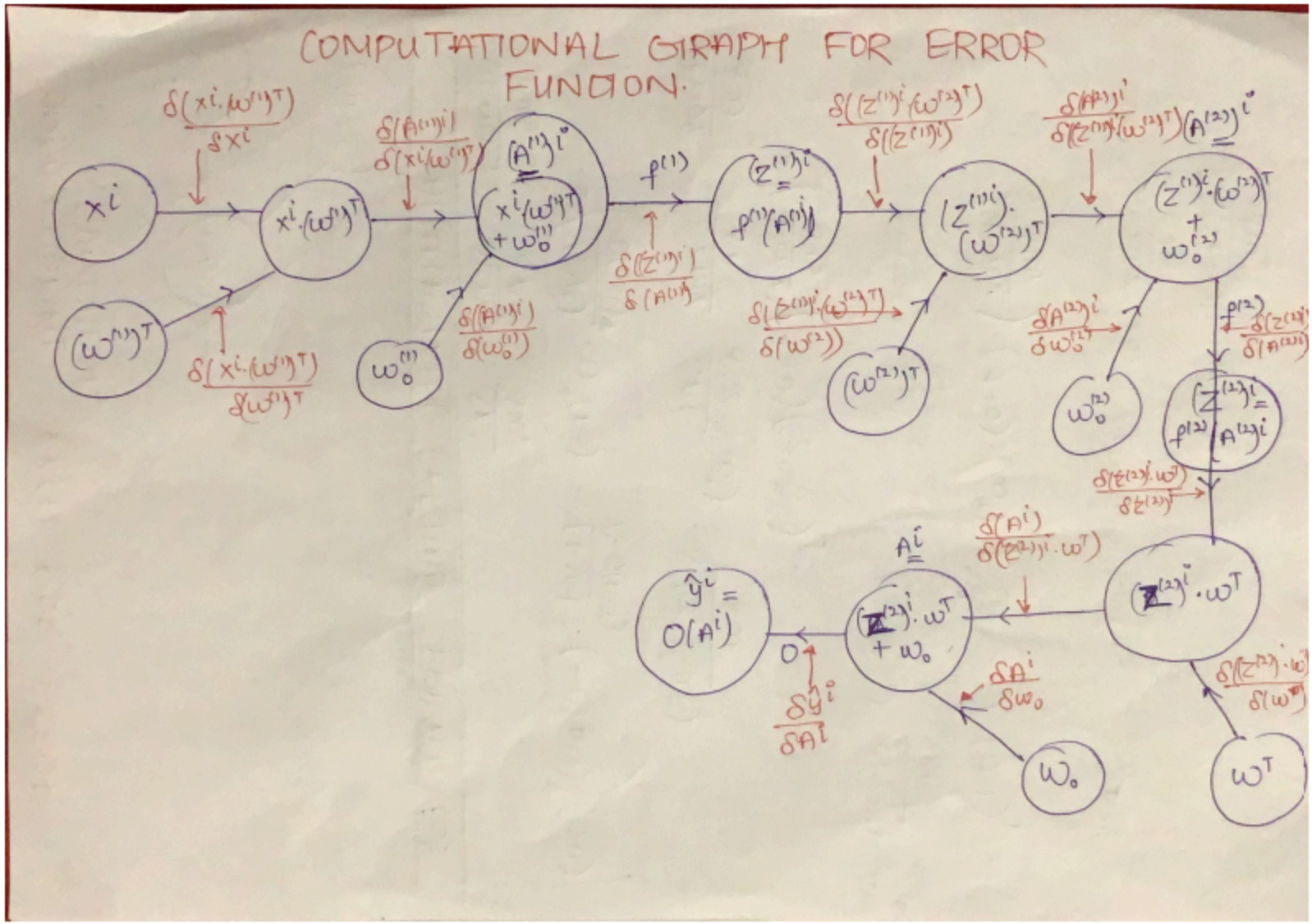
$\hat{y}^i = [\hat{y}_1^i \ \hat{y}_2^i \ldots \hat{y}_K^i] = [O(A_1)^i \ldots O(A_K)^i]$

$= O[A_1^i \ldots A_K^i]$

$= O(A^i)$

Stochastic Error. $= -y^i \ln(O(A^i)^T) - (1-y^i)\ln(1-(O(A^i))^T)$

COMPUTATIONAL GRAPH FOR ERROR FUNCTION.



CALCULATING PARTIAL DERIVATIVES.

$$\frac{\delta Error}{\delta w_0} = \frac{\delta Error}{\delta A^i} \cdot \frac{\delta A^i}{\delta w_0}$$

$$= \left(\frac{(-y^i)}{\sigma(A^i)} \cdot \sigma'(A^i) - \frac{(1-y^i)}{(1-\sigma(A^i))} \cdot (-\sigma'(A^i)) \right) (0+1)$$

$$\frac{\delta Error}{\delta w} = \frac{\delta Error}{\delta A^i} \cdot \frac{\delta A^i}{\delta ((Z^{(2)})^i \cdot w^T)} \cdot \frac{\delta ((Z^{(2)})^i \cdot w^T)}{\delta w}$$

$$= \left(\frac{\delta Error}{\delta w_0} \right) (1+0) \left((Z^{(2)})^i \right)$$

$$\frac{\delta Error}{\delta w_0^{(2)}} = \frac{\delta Error}{\delta A^i} \cdot \frac{\delta A^i}{\delta ((Z^{(2)})^i \cdot w^T)} \cdot \frac{\delta ((Z^{(2)})^i \cdot w^T)}{\delta (Z^{(2)})^i} \cdot \frac{\delta (Z^{(2)})^i}{\delta (A^{(2)})^i} \cdot \frac{\delta (A^{(2)})^i}{\delta w_0^{(2)}}$$

$$= \left(\frac{\delta Error}{\delta w_0} \right) \cdot (1+0) \cdot (w^T) \cdot \left(f^{(2)'}((A^{(2)})^i) \cdot 1 \right) \cdot (0+1)$$

$$\frac{\delta Error}{\delta w^{(2)}} = \frac{\delta Error}{\delta w_0} \cdot (w^T) \cdot \left(f^{(2)'}((A^{(2)})^i) \cdot 1 \right) (1+0) \cdot ((Z^{(1)})^i)$$

$$\left(\frac{\delta Error}{\delta w_0^{(2)}} \right) \cdot ((Z^{(1)})^i)$$

$$\frac{\delta Error}{\delta w_0^{(1)}} = \frac{\delta Error}{\delta w_0} \cdot (w^T) \cdot \left(f^{(2)'}((A^{(2)})^i) \cdot 1 \right) \cdot (1+0) \cdot (w^{(2)})^T \cdot \left(f^{(1)'}((A^{(1)})^i) \cdot 1 \right) \cdot (0+1)$$

$$\frac{\delta Error}{\delta w^{(1)}} = \left(\frac{\delta Error}{\delta w_0^{(1)}} \right) \cdot (1+0) \cdot (x^i)$$