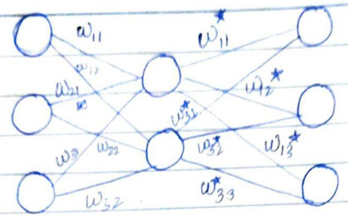
Assignment number:4
Name: Manasvi Singh
Roll number:2019369

Question A)
In part 2, the input layer will have three nodes, the hidden layer will have 2 nodes and the output layer will have 3 nodes(as autoencoder)
I have used matrices to keep track of Weights and biases,
The derivation for backpropagation is done down below:

Ques 2 Part C .



$$W1 = \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \\ w_{31} & w_{32} \end{bmatrix} \cancel{w_{13}}$$

Input in hidden layer .

$w_1^T x + B_1$.

$$\begin{bmatrix} w_{11} & w_{21} & w_{31} \\ w_{12} & w_{22} & w_{32} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ \cancel{b_3} \end{bmatrix}$$

$$I_1 = \begin{bmatrix} w_{11} x_1 + w_{21} x_2 + w_{31} x_3 + b_1 \\ w_{12} x_1 + w_{22} x_2 + w_{32} x_3 + b_2 \end{bmatrix} \Bigg\} \text{Input to hidden layer .}$$

$$\frac{\partial E}{\partial W1} = \overbrace{\underbrace{\frac{\partial L}{\partial O2} \frac{\partial O2}{\partial I2}}_{\text{found in previous step.}} \frac{\partial I2}{\partial O1} \frac{\partial O1}{\partial I1}}^{\alpha} \frac{\partial I1}{\partial W1}$$

$I2 = W2O1 + B2$

$\frac{\partial I2}{\partial O1} = W2.$ ↳ Weight matrix of second layer.

$\frac{\partial O1}{\partial I1} = \sigma'(I1)$ $\because O1 = \sigma(I1)$

$\frac{\partial I1}{\partial W1} = x$ ↳ Input.

$\underset{\substack{\downarrow \\ \text{Weight matrix of first layer}}}{w_1} x + B_1$

$$\underbrace{\frac{\partial E}{\partial B2}}_{\text{Bias b/w hidden \& output layer.}} = \underbrace{\frac{\partial L}{\partial O2} \frac{\partial O2}{\partial I2}}_{\text{Calculated}} \frac{\partial I2}{\partial B2}$$

$\frac{\partial I2}{\partial B2}$ ↳ I ↳ ~~Indentity~~ matrix of B2 dimension having all 1.

$$\frac{\partial E}{\partial B1} = \alpha \times \underbrace{\frac{\partial I1}{\partial B1}}$$

↳ matrix of B1 dimension with all 1.

Output of Hidden layer

$$O1 = \sigma(I_1) = \begin{bmatrix} \sigma(w_{11}x_1 + w_{21}x_2 + w_{31}x_3 + b_1) \\ \sigma(w_{12}x_1 + w_{22}x_2 + w_{32}x_3 + b_2) \end{bmatrix}$$

$\hookrightarrow \begin{bmatrix} O_1 \\ O_2 \end{bmatrix}$

Input to Output layer $\longrightarrow w^{*T}O_1 + B_2 = I2$

O_2 = Output of Output layer $= \sigma(\underbrace{(w^*)^T}_{3\times 2}O_1 + B_2)$ $\quad 2\,1$

$$w^* = \begin{bmatrix} w^*_{11} & w^*_{12} & w^*_{13} \\ w^*_{21} & w^*_{22} & w^*_{23} \end{bmatrix} \qquad B_2 = \begin{bmatrix} b_1^* \\ b_2^* \\ b_3^* \end{bmatrix}$$

Upon Backpropagation $\quad W2 = w^*$

$E = (\hat{x} - x)^2$.

$O2 = \sigma(I2)$

$$\frac{\partial E}{\partial W2} = \frac{\partial E}{\partial O2}\,\frac{\partial O2}{\partial I2}\,\frac{\partial I2}{\partial W2}$$

$\hookrightarrow 2(\hat{x} - x)$

$$\frac{\partial O2}{\partial I2} = \sigma'(I2) \qquad \frac{\partial I2}{\partial W2} = O_1$$

Updation Rule.

$$W1\_new = W1 - \text{learning rate} * \frac{\partial E}{\partial W1}$$

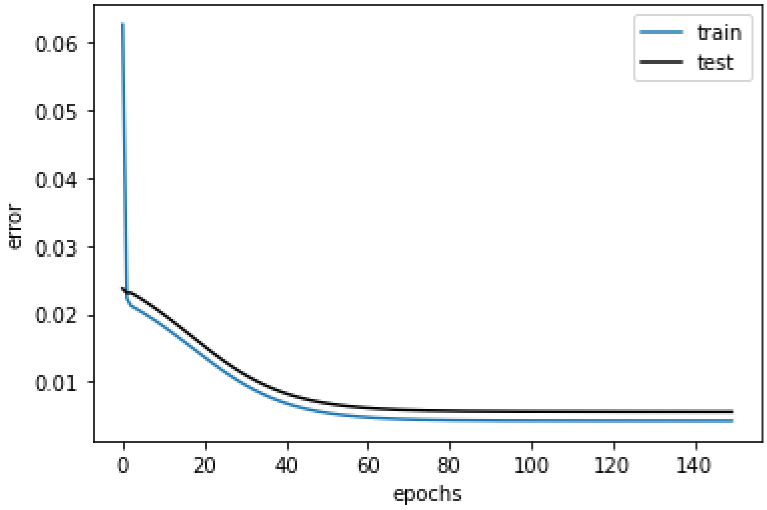
$$W2\_new = W2 - \text{learning\_rate}\ \frac{\partial E}{\partial W2}.$$

$$B1\_new = B1 - \text{learning-rate}\ \frac{\partial E}{\partial B1}$$

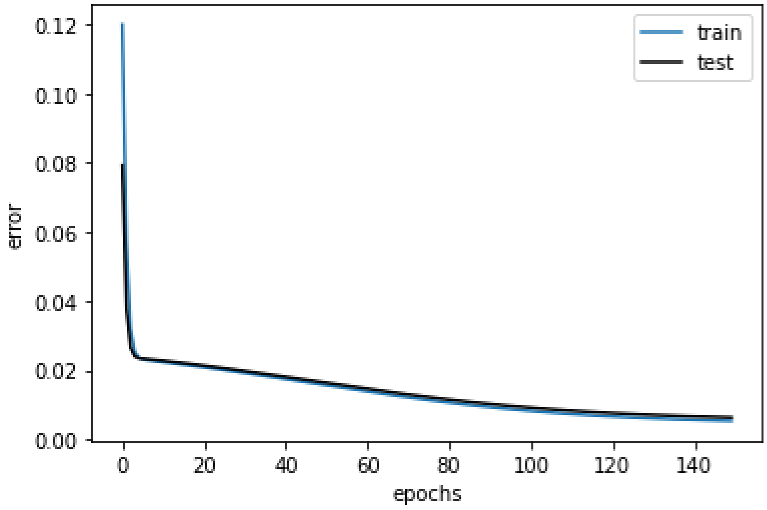
$$B2\_new = B2 - \text{learning-rate}\ \frac{\partial E}{\partial B2}.$$

Question 3)

Backpropagation which is implemented from scratch



For the autograd part:



As can be seen from both graphs that error decreases as epochs increase.

Question 4)For lesser number of epochs,backpropagation shows better learning for training dataset by giving less error,whereas it takes a little more time for aurograd.But for testing dataset autograd performs better and gives lesser error as can be seen for the graph.

Even for a larger number of epochs,autograd performs better for testing dataset.
The error during the initial epochs is higher for autograd as compared o bakpropgation.

Also for autograd,the graphs for testing and training coincide after some number of epochs whereas in backpropgation implemented from scratch it does not and testing error remains higher as compared to training dataset after only some epochs.

The initial training error is high for both as random weights are assigned at starting and that leads to a big error.