Q1

a) Regression is ~~are~~ used to ~~[illegible]~~ predict a value ~~from~~ using data already provided to us by the user. With the given data, our model makes adjustments in values of parameter (randomly ~~β~~ set at first) such that the squared error is least and comes up with an equation of curve which helps in our prediction.

- Minimizing sq. error helps us computationally. ~~Diffra~~
- It makes the function differentiable, making it easier to find grad. of cost fun^c
- It also makes all values ~~[illegible]~~ of error positive. Direct summation ~~[illegible]~~ may lead to zero error, telling our model has 100% accuracy, which isn't true.

b) $J = \frac{1}{2}\|y - X\beta\|^2$

$= \frac{1}{2}(y - X\beta)^T(y - X\beta)$

$y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}_{n\times 1}$

$X = \begin{bmatrix} \quad \end{bmatrix}_{n\times(d+1)}$

$\nabla(J) = \nabla\left(\frac{1}{2}\left(y^Ty - y^TX\beta - \beta^TX^Ty + \beta^TX^TX\beta\right)\right)$

$\beta = \begin{bmatrix} \beta_0 \\ \vdots \\ \beta_d \end{bmatrix}_{(d+1)\times 1}$

$y^TX\beta \rightarrow (1\times n)(n\times(d+1))((d+1)\times 1)$

$\Rightarrow (1\times 1)$

$\hookrightarrow$ Scalar

$$\nabla\left(\frac{1}{2}\left(y^Ty - y^TX\beta - \underbrace{(y^TX\beta)^T}_{\text{Scalar } \therefore \text{ equal.}} + \beta^TX^TX\beta\right)\right)$$

$= \nabla\left(\frac{1}{2}\left(y^Ty - 2\beta^TX^Ty + \beta^TX^TX\beta\right)\right)$

$= \frac{1}{2}\left(-X^Ty + X^TX\beta\right) = 0$

Page No.
Date

$\Rightarrow X^T y = X^T X \beta$

~~$y = x$~~

$\Rightarrow \beta = \underline{(X^T X)^{-1}} X^T y$

c) Multiplication and taking inverse of a high dimensional matrix is very time taking and also computationally expensive.
In gradient descent, ~~we don't ... quite require~~ we don't have to calculate inverse of a matrix, this makes the process very fast, also at the same time taking less memory.

Q2
a) In back propogation, we compute the gradient of our loss func and adjust our parameters and biases accordingly so that our loss func minimises.
Chain Rule makes our computation cost low ~~as~~ as it is much easier and faster.

b) $L = -y \log a_2 + (1-y)\log(1-a_2)$

$\frac{\partial L}{\partial z_2} = a_2 - y$ $\qquad z_2 = w_2 a_1 + b_2$

$\frac{\partial L}{\partial w_2} = \frac{\partial L}{\partial z_2} \frac{\partial z_2}{\partial w_2}$

$= (a_2 - y)(a_1)$

$\frac{\partial L}{\partial b_2} =$ ~~$\frac{\partial L}{\partial w_2} \; \partial z_2$~~ $\frac{\partial L}{\partial z_2} \frac{\partial z_2}{\partial b_2}$

$= (a_2 - y)(1)$

~~$\frac{\partial L}{\partial w_1} = \left(\frac{\partial L}{\partial z_2} \frac{\partial z_2}{\partial w_1} \right) \frac{\partial z_2}{\partial a_1} \frac{\partial a_1}{\partial w_1}$~~

~~$= (a_2 - y)(w_2)(a_1)(1 - a_1)$~~

~~$\frac{\partial L}{\partial b_1} = \left(\frac{\partial L}{\partial z_2} \right. \partial z_2$~~

$\frac{\partial L}{\partial w_1} = \frac{\partial L}{\partial z_1} \frac{\partial z_1}{\partial w_1}$

$= \frac{\partial L}{\partial z_2} \frac{\partial z_2}{\partial a_1} \frac{\partial a_1}{\partial z_1} \frac{\partial z_1}{\partial w_1}$

$= (a_2 - y) w_2 a_1 (1 - a_1) x$

$\frac{\partial L}{\partial b_1} = \frac{\partial L}{\partial z_1} \frac{\partial z_1}{\partial b_1}$

$= (a_2 - y) w_2 a_1 (1 - a_1)(1)$

c) $w_1' := w_1 - \alpha \frac{\partial L}{\partial w_1}$

$w_2' := w_2 - \alpha \frac{\partial L}{\partial w_2}$

$b_1' := b_1 - \alpha \frac{\partial L}{\partial b_1}$

$b_2' := b_2 - \alpha \frac{\partial L}{\partial b_2}$

α (learning rate) Controls step size ie, if the step is too large, it may completely miss the ~~local~~ minima or if the step is too small it converges very slowly.

Page No.
Date

Q3

~~a) In ANN, the input is processed independently whereas in RNN is designed for sequential data.~~

~~RNN have a hidden layer, whereas ANN doesn't~~

a) RNNs process input sequentially, by using an hidden state that serves as a memory of past inputs.

ANN processes input by passing data through layers of interconnected neurons..

b) RNN struggle with long term dependencies due to vanishing grad. problem

c) Role of gates in LSTMs is ~~[illegible]~~ to control flow of information. which information to forget / input / output.
The gates allow LSTMs to retain selective information, helping in long term memory

d) With the help of gates in LSTMs, gradients vanishes very slowly addressing to the problem of vanishing grad.

e) ANN → Price of house prediction
RNN → Word predictor / Sentence Completor.
LSTM → Language translation.

Page No.
Date

Q4

Eg: The ~~dog~~ man asked his girlfriend to marry him.

To correctly predict the last word (him) the model must remember that we are talking about ~~them~~ the man and not his girlfriend.

Yes, a standard RNN would struggle because of the vanishing grad., as a result the model will forget about the man ~~as~~ by the time it reaches to '~~him~~' 'marry'.

b) The LSTM gates allow LSTMs to retain relevant info. for long time.
The gates, forget gate, input gate, output gate, help in this ~~tasks~~ tasks.
It forgets the irrelevant info and only retains the important info.

Eg ~~I went to store as~~ I went for a walk as I needed some fresh air.

The model must remember that the ~~s~~ sencentence is talking about me to predict the ~~as~~ second I in the sentence.