Date __/__/____

Assignment 1

1.(a) Linear Regression tells relationship between input features x and output y in linear function. Minimizing squared error function gives stable solution ~~by~~ because it is smooth and differentiable and penalizes large errors more than smaller ones.

$\hat{y}_0 = X\beta$ (Linear Regression)

$\frac{1}{2}(y_i - \hat{y}_i)^2$ (Squared Error Loss)

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

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

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

$\hookrightarrow J_\beta = -X^T(y - X\beta)$ (w.r.t β)

$X^T(y - X\beta) = 0$ (gradient to 0)

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

$\beta = (X^T X)^{-1} X^T y$ (normal equation)

(c) Direct inversions of $X^T X$ in normal equation may be problematic because -

- It needs a lot of memory & computation when dataset is large
- It does not work well for very large dimensional data
- Small calculation errors can become big, so that result is not reliable
- Matrix inversion takes a lot of time

Date ___/___/_____

Iterative methods are usually preferred because
→ no matrix inversion
→ works with small batches
→ Memory efficient
→ Scales to large datasets

2. (a) Backpropagation is a method that helps a neural network learn from its mistakes.
Forward pass: Network makes a prediction using the current weights
Backward pass: checks how wrong the prediction is and sends error backward
Using this, network slightly changes weights so that next prediction is better.

As the output depends on many layers, each layer depends on previous one so the loss is not direct function it is chain of functions.
"ChainRule" lets us break complex derivatives into small, simple derivatives

(b) $z_1 = w_1 x + b_1 \qquad a_1 = \sigma(z_1)$
$z_2 = w_2 a_1 + b_2 \qquad a_2 = \hat{y} = \sigma(z_2)$
$L = -[y \log a_2 + (1-y) \log(1-a_2)]$ } given

Date ___/___/_____

→ $\frac{\partial L}{\partial w_2} = \frac{\partial z_2}{\partial w_2} \frac{\partial L}{\partial z_2}$ (by chain Rule)

$\boxed{\frac{\partial L}{\partial w_2} = a_1 (a_2 - y)}$ $\left(z_2 = w_2 a_1 + b_1, \quad \frac{dz_2}{\partial w_2} = a_1 \right)$

$\frac{\partial L}{\partial b_2} = \frac{\partial z_2}{\partial b_2} \frac{\partial L}{\partial z_2}$ (chain Rule)

$= 1 \times L$

$\frac{\partial L}{\partial b_2} = \frac{\partial z_2}{\partial b_2} \frac{\partial L}{\partial z_2}$ (chain Rule)

$\boxed{\frac{\partial L}{\partial b_2} = 1 \ (a_2 - y)}$ $\left(z_2 = w_2 a_1 + b_2, \quad \frac{dz_2}{\partial b_2} = 1 \right)$

→ $\frac{\partial L}{\partial w_1} = \frac{\partial z_2}{\partial w_1} \frac{\partial L}{\partial z_2}$ (Chain Rule)

$\boxed{\frac{\partial L}{\partial w_1} = w_2 x \, a_1 (1 - a_1) (a_2 - y)}$

$z_2 = w_2 a_1 + b_2$

$a_1 = \frac{1}{1 + e^{-(w_1 x + b_1)}}$

$z_2 = \frac{w_2}{1 + e^{-(w_1 x + b_1)}} + b_2$

$\frac{dz_2}{\partial w_1} = \frac{w_2 (-x) e^{-(w_1 x + b_1)}}{(1 + e^{-(w_1 x + b_1)})^2}$

$= + w_2 a_1 (1 - a_1) x$

→ $\frac{\partial L}{\partial b_1} = \frac{\partial z_2}{\partial b_1} \frac{\partial L}{\partial z_2}$ (Chain Rule)

$z_2 = w_2 a_1 + b_2$

$a_1 = \frac{1}{1 + e^{-(w_1 x + b_1)}}$

$z_2 = \frac{w_2}{1 + e^{-(w_1 x + b_1)}} + b_2$

Date ___/___/_____

$$\frac{\partial z_2}{\partial b_1} = -w_2 \frac{e^{-(w_1 x + b_1)}}{(1+e^{-w_1 x + b_1})^2}$$

$$\frac{\partial z_2}{\partial b_1} = w_2 a_1 (1-a_1)$$

$$\# \quad \boxed{\frac{\partial L}{\partial b_1} = w_2 a_1 (1-a_1)(a_2 - y)}$$

(c) $w_1 \rightarrow w_1 - \eta \frac{\partial L}{\partial w_1}$

$w_2 \rightarrow w_2 - \eta \frac{\partial L}{\partial w_2}$

$b_1 \rightarrow b_1 - \eta \frac{\partial L}{\partial b_1}$

$b_2 \rightarrow b_2 - \eta \frac{\partial L}{\partial b_2}$

where use these equations to update w_1, w_2, b_1 & b_2 in gradient descent.

η = learning rate ; it controls how big or small step to take while updating parameters.
large $\eta \Rightarrow$ fast learning but may lead to unstable learning
small $\eta \Rightarrow$ slow learning but stable.

3 (a) ANN (Artificial Neural Network) processes inputs independently and has no memory of past inputs whereas RNN (Recurrent Neural Network) is designed for sequence data with memory of past inputs.

Page No. ____

Date __/__/____

(b) Simple RNN struggles with long term dependencies as it suffers from vanishing gradients and cannot remember long term ~~dependence~~ dependencies

(c) Role of gates in LSTMs are
→ forget Gate → It decides what information to remove from memory
→ Input gate → decides what new information to add
→ Output gate → decides what information to pass next step.

Gates helps to preserve long term information as memory cell carries information forward with very little change. gates control updates smoothly instead of overwriting

(d) LSTMs fix vanishing gradients as memory cell allows constant error flow and gates regulate information flow. In LSTM,

(e) Example Tasks;
ANN → House price prediction
RNN → ~~language~~ speech Recognition
LSTM → Machine Translation

4.(a) "Manav met Dev after the meeting, and he was exhausted"

The word 'he' is a pronoun, and to understand correctly, we must know 'he' refers to — Manav or Dev.

This creates long term dependency because the meaning of 'he' depends on information that occured several steps earlier.

Yes, RNN would struggle ~~to~~ as in a standard RNN, information about earlier words gradually fades due to vanishing gradient problem. By the time "he" it may no longer retain clear information about Manav or Dev. This can lead to incorrect or ambigous pronoun Resolution.

(b) The LSTM stores entity information (Manav or Dev) in its memory cell while reading the sentence. This information is carried forward across multiple words.
While processing intermediate words, the forget gate remains close to 1 and prevents loss of entity information.
If the forget gate were close to 0 too early, the entities would have been forgotten.