Date: __/__/____

## Theory

Answer 1

a) We have a classification problem where we have to label papaya

$$f(x_i) \quad \begin{cases} \text{sweet} = 1 & x_i = 1 \\ \text{not} = 0 & x_i = 0 \end{cases}$$

We want to maximize the Likelihood function of $f(x, \theta)$ : function to correctly predict the label

$$L(x_1, x_2 \dots x_n) = P(x_1 = x_1, x_2 = x_2 \dots, x_n = x_n \mid y_1 = y_1, y_2 = y_2 \dots, \theta)$$

$$= \prod_{y_i} f(x_i) \cdot \prod_{1-y_i} 1 - f(x_i)$$

$$L(\theta) = \prod f(x_i)^{y_i} (1 - f(x_i))^{1-y_i}$$

we need to find $\theta$ parameter that maximizes likelihood function

which is BINARY CROSS ENTROPY function

MSE is NOT a good option because :

1) MSE assumes error to be a Normal distribution, But in the classification problem error can take only 2 values $\{0, 1\}$ and is a Bernoulli R.V. So, use of MSE doesn't make sense.

Date: __/__/____

2) Non Linear Relationship: MSE is a linear function while use of sigmoid function make the model's output non linear

3) Poor Indication of Misclassification: MSE doesn't provide clear indication of how many instances were misclassified, it focuses on magnitude of error.

4) Gradient Vanishing Problem

b) BCE: Binary Cross Entropy Loss

$$L(y, \hat{y}) = -\sum_{c=1}^{M} y_{o,c} \log(\hat{y}_{o,c})$$

For binary cross entropy:

$$L(y, \hat{y}) = -(y_i \log \hat{y}_i + (1-y_i) \log(1-\hat{y}_i))$$

c) $\hat{y} = 0.9$

$$L(y, \hat{y}) = -(0 \log 0.9 + 1 \log 0.1)$$

$$= -\log 0.1 = +2.303$$

d) $y = [1\ 0\ 0]$

$\hat{y} = [0.1\ 0.2\ 0.7]$

$$L(y, \hat{y}) = -\left(\sum_{i=0}^{n} y_i \log \hat{y}_i\right) \qquad L = \frac{BCE(y_1, \hat{y}) + BCE(y_2, \hat{y}) + BCE(}{3}$$

$$= -(1 \log 0.1 + 1 \log 0.8 + 1 \log 0.3 + 1 \log 0.2 + 1 \log 0.7)$$

$$= -\log 0.1$$

$$= +2.303 + 0.2231 + 1.2041 \approx \boxed{1.24}$$

#learnthesmarterway

Date: __/__/____

d) contd.

1st sample : $BCE(y_1, \hat{y}_1) \approx 2.3026$

2nd sample : $BCE(y_2, \hat{y}) \approx 0.2231$

3rd sample : $BCE(y_3, \hat{y}) = 1.2041$

$$L = \frac{BCE(y_1, \hat{y}_1) + BCE(y_2, \hat{y}) + BCE(y_3, \hat{y})}{3}$$

$$L \approx \frac{2.3026 + 0.2231 + 1.2041}{3} \approx 1.24$$

c) $W_{new} \leftarrow W - \alpha \left( \frac{\partial L_{BCE}}{\partial W} + \lambda W \right)$

We penalise big weights so as to prevent overfitting. In $L_2$ we penalize it with $\frac{\partial L_{BCE}}{\partial W} + \lambda W$

If I have 2 model's A & B

A uses $L_2$ regularization
B does NOT

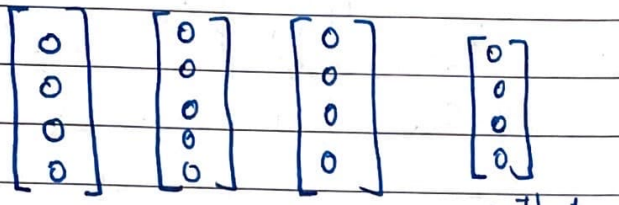
any weight w that is larger than the approximate central tendancy would be penalized in A

Hence weights in A would **generally be smaller** than B

Exception: if W is -ve
then $\lambda W + \frac{\partial L_{BCE}}{\partial W}$ term may also be negative and in that case W is pushed towards 0
Hence, in the case when W is -ve,
$W_B > W_a$ $W_a < W_b$

Date: __/__/____

Ans 2




$x$ : $D_x \times 1$

$z^1 = W^1 x + b^1$

$a^1 = \text{LeakyReLU}(z^1, \alpha = 0.01)$

$z^2 = W^2 a^1 + b^2$

$\hat{y} = \text{softmax}(z^2)$

$L = -\sum y_i \log \hat{y}_i$

a) $z^1 = W^1 x + b^1$

$D_a \times 1 \quad D_m \times 1 + D_a \times 1$

$\downarrow$

$D_a \times D_x$

shape of $W^1$ : $D_a \times D_x$

$b^1 = D_a \times 1$

$a^1 = D_a \times 1$

$z^2 = W^2 a^1 + b^2$

$K \times 1 \quad D_a \times 1 \quad K \times 1$

Date: __/__/____

~~$W^2$ : $dk \times$~~

$W^2$ : $K \times D_a$

$b^2$ : $K \times 1$

if hidden layer if batch size is $m$

$Z^{[1]}$ : $D_a \times m$

b) We have got $\hat{y}$ : which is a $K \times 1$ matrix

$y$ : one hot encoding vector of $K \times 1$ matrix

We calculate Loss as

$$L = -\sum_i y_i \log(\hat{y}_i)$$

and backpropogate loss

$$\hat{y}_i = \frac{e^{z^2_i}}{\sum_{i=1}^{K} e^{z^2_i}} \qquad \hat{y}_k = \frac{e^{z^2_k}}{\sum_{i=1}^{K} e^{z^2_i}}$$

$$= -(1 - y_k)$$

$$\frac{\partial \hat{y}_k}{\partial z^{[2]}_k} = \sigma(z^{[2]}_k)\,(1 - \sigma(z^{[2]}_k)) = y_k \times (1 - y_k)$$

$\sigma(x)$ : signifies sigmoid fn of on $x$

c) $$\frac{\partial \hat{y}_k}{\partial z^{[2]}_i} = \sigma(z^{[2]}_i)\,(1 - \sigma(z^{[2]}_{k\,i}))$$

$$= y_i(1 - y_i) \qquad -y_i y_k$$

Date: __/__/____

d) $L = -\sum y_i \log \hat{y}_i$

$$\frac{\partial L}{\partial y} = \begin{bmatrix} \frac{\partial L}{\partial y_1} & \frac{\partial L}{\partial y_2} & \cdots & \frac{\partial L}{\partial y_n} \end{bmatrix}$$

$$\frac{\partial L}{\partial z_i^{[2]}} = \frac{\partial L}{\partial y} \cdot \frac{\partial y}{\partial z_i^{[2]}}$$

$$\sum_{j=1}^{n} \frac{\partial L}{\partial y_j} \cdot \frac{\partial y_j}{\partial z_i^{[2]}}$$

$$\frac{\partial L}{\partial z_i^{[2]}} = \frac{\partial L}{\partial y_k} \cdot \frac{\partial y_k}{\partial z_i^{[2]}}$$ (rest all terms are 0 because of 1 hot encoding vector)

if $i == k$: $\frac{\partial L}{\partial z_i^{[2]}} = \frac{-1}{y_k} \cdot y_k (1 - y_k) = y_k - 1 = y_i - 1$ $(k = i)$

else if $i != k$: $\frac{\partial L}{\partial z_i^{[2]}} = \frac{-1}{y_k} \cdot y_k \cdot y_i = -y_i$

e) softmax function : $\frac{e^{x_i}}{\sum_{i \in N} e^{x_i}}$

since softmax exponentiates $x_i$, for large $x$ $e^{x_i}$ will be a really big value and hard to compute

Date: __/__/____

Hence softmax function is numerically UNSTABLE

A modification in softmax function is done

$$\frac{e^{x_i - z}}{\sum e^{x_i - z}}$$

where z val is the max $(x_i)$ $\forall i$

this transforms all x into values between [0, 1] and normalises and provides softmax output