Homework 2

Problem 1:

Homework - 2

Q.1]

→ For XOR problems, $X = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix}$, $\omega = \begin{bmatrix} \omega_1 \\ \omega_2 \end{bmatrix}$

The MSE loss functions is, $f_{mse} = \frac{1}{4} \sum_{x \in X} (\hat{y} - \hat{y})^2$

where $\hat{y} = x^T \omega + b$

consider $x = [x_1 \; x_2]^T$

so, $\hat{y} = \omega_1 x_1 + \omega_2 x_2 + b$

$$f_{mse}(\omega, b) = \frac{1}{4} \sum_{i=1}^{4} \left[(\omega_1 x_1^i + \omega_2 x_2^i + b) - y^i \right]^2$$

We know that,
1. $(0,0) \rightarrow y = 0$
2. $(0,1) \rightarrow y = 1$
3. $(1,0) \rightarrow y = 1$
4. $(1,1) \rightarrow y = 0$

$$\therefore f_{mse}(\omega, b) = \frac{1}{4} \left[(b-0)^2 + (\omega_2 + b - 1)^2 + (\omega_1 + b - 1)^2 + (\omega_1 + \omega_2 + b - 0)^2 \right]$$

Now, derive f_{mse} w.r.t ω_1, ω_2 and b and set them to 0.

$$\frac{\partial f}{\partial \omega_1} = \frac{1}{4} \left[2(\omega_1 + b - 1) + 2(\omega_1 + \omega_2 + b) \right]$$

$$0 = \frac{1}{4} \left[4\omega_1 + 4b + 2\omega_2 - 2 \right]$$

$$\rightarrow \omega_1 + b + \tfrac{1}{2}\omega_2 - \tfrac{1}{2} = 0 \qquad -(1)$$

$$\frac{\partial f}{\partial \omega_2} = \frac{1}{4} \left[2(\omega_2 + b - 1) + 2(\omega_1 + \omega_2 + b) \right]$$

$$0 = \frac{1}{4} \left[4\omega_2 + 4b + 2\omega_1 - 2 \right]$$

$\rightarrow \quad w_2 + b + \frac{1}{2} w_1 - \frac{1}{2} = 0 \qquad - (2)$

$$\frac{\partial f}{\partial b} = \frac{1}{4}\left[2b + 2(w_1+b-1) + 2(w_2+b-1) + 2(w_1+w_2+b) \right]$$

$$0 = \frac{1}{4}\left[8b + 4w_1 + 4w_2 - 4 \right]$$

$\rightarrow \quad 2b + w_1 + w_2 - 1 = 0 \qquad - (3)$

Now, solving for (1) and (2), if we subtract them we get $\boxed{w_1 = w_2}$

Now apply this to equation (1)

$$w_1 + b + \frac{1}{2} w_1 = \frac{1}{2}$$

$$\frac{3}{2} w_1 + b = \frac{1}{2}$$

Now, If we substitute $w_1 = 0$ then, $b = \frac{1}{2}$ and $w_1 = w_2 = 0$

$\therefore$ After minimizing MSE function, we get $w_1 = 0$, $w_2 = 0$, $b = 0.5$
So, for all values of x, the best prediction value $\hat{y}$ will be 0.5.

Hence, 2-layer NN cannot learn the XOR function.

Problem 2

Q2) $\hat{y}_k = \frac{\exp z_k}{\sum_{k'=1}^{c} \exp z_{k'}}$, $z_k = x^T w_k + b_k$ for each $k = 1, 2, \dots c$

$f_{ce}(w,b) = \frac{-1}{n} \sum_{i=1}^{n} \sum_{k=1}^{c} y_k^i \log \hat{y}_k^i$

Now, find gradient of cross entropy wrt each weight vector w_L, where $L \in \{1, 2, \dots, c\}$

$\frac{\partial f_{ce}}{\partial w_L} = \frac{-1}{n} \sum_{i=1}^{n} \sum_{k=1}^{c} y_k^i \frac{\partial}{\partial w_L} \log \hat{y}_k^i$

As, $\frac{\partial}{\partial w_L} \log \hat{y}_k^i = \frac{1}{\hat{y}_k^i} \frac{\partial \hat{y}_k^i}{\partial w_L}$, using Chain Rule

Now, we have to handle 2 cases:

1] $L = k$ (for correct class)

2] $L \neq k$ (for incorrect class)

For case 1, $L = k$

$$\frac{\partial \hat{y}_L^i}{\partial w_L} = x^i \hat{y}_L^i (1-\hat{y}_L^i)$$

As correct class derivation involves product of x^i, $\hat{y}_L^i$ and $(1-\hat{y}_L^i)$ for softmax normalization.

For case 2, $L \neq k$

$$\frac{\partial \hat{y}_L^i}{\partial w_L} = -x^i \hat{y}_k^i \hat{y}_L^i$$

As, −ve sign comes from changing the weight of class L decrease its probability for class k.

Now when we combine both cases to get total gradient we have:

$$\frac{\partial f_{CE}(W,b)}{\partial w_L} = \frac{-1}{n} \sum_{i=1}^{n} x^i (y_L^i - \hat{y}_L^i)$$

Similarly, we can derive ~~grad~~ wrt $b^{(L)}$, without x^i since b is not multiplied by x.

$$\therefore \nabla_b f_{CE}(W,b) = \frac{-1}{n} \sum_{i=1}^{n} (y^i - \hat{y}^i)$$

$\therefore$ gradient updates for softmax are :-

$$w_L \leftarrow w_L - \varepsilon \nabla_{w_L} f_{CE}(W,b)$$

$$b_L \leftarrow b_L - \varepsilon \nabla_b f_{CE}(W,b)$$

Where, $\nabla_{(b,w)} f_{CE}(W,b) = \frac{-1}{n} \sum_{i=1}^{n} (y^i - \hat{y}^i)$

Problem 4

Q.4]

→ $\hat{y} = \sigma(x^T w + b)$

We train this function using log loss

a) For a well-chosen learning rate.

All the training examples are positively labeled, then the $\hat{y}$ (the predicted output) will be 1 for every example

$$\text{Log Loss}, f = \log(\hat{y})$$

So, if $\hat{y} = 1 \rightarrow \log(\hat{y}) = 0$

Therefore, loss converges to 0 with a well-chosen learning rate.

b) The convergence of bias b, depends on the training examples

If data is linearly separable, the bias term can adjust itself so that decision boundary shifts to fit the data.

If data is not linearly separable, bias 'b' will not converge as the training model will find it hard to separate the examples c)

→ consider 2 examples $x_1, x_2 \in R^2$

$x_1 = [2 \ \ 0]^T$, $x_2 = [0 \ \ 2]^T$

positive negative

These points can be separated, for a well-chosen learning rate 'w' will adjust and converge

And if $x_1 = [2 \ \ 2]^T$ $x_2 = [4 \ \ 4]^T$

These points lie on same line, so ~~for~~ even for a well-chosen learning rate the gradient will fluctuate and w will not converge anyhow.