ASSIGNMENT → 2

NAME :- PUSHPENDRA SINGH CHAUHAN
STUDENT ID :- A20472647
COURSE NO. :- CS577
SEMESTER :- SPRING 2021

Artificial Neurons

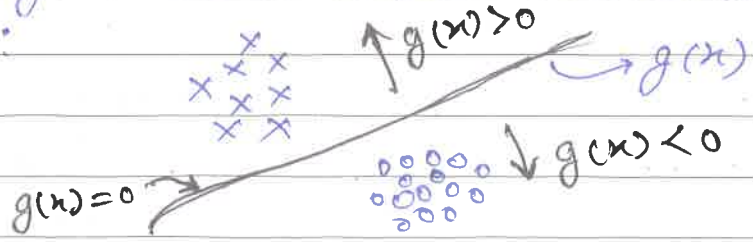
Q.1). Given a neuron with linear activation and weights (0.1, 0.2, 0.3). Compute the output for the input vector (1, 1).

Sol:- Output for the input vector (1, 1)

$= 0.2 \times 1 + 0.3 \times 1 + 0.1$
$= 0.2 + 0.3 + 0.1$
$= 0.6$

Q.2). Given a linear discriminant function $g(x)$ explain what should be its value on one side of the decision boundary, on the other side of the decision boundary, and on the decision boundary.

Sol:- Let $g(x)$ be a linear discriminant function, then:



So,

→ $g(x) > 0$ on one side of the decision

boundary.

→ $g(x) < 0$ on the other side of the decision boundary.

→ $g(x) = 0$ on the decision boundary.

Q.3) Given a linear discriminant function $g(x_1, x_2) = \theta_0 + \theta_1 x_1 + \theta_2 x_2$. Explain the meaning of the coefficients $\theta_0, \theta_1, \theta_2$.

Soln:- (a). Here, for geometric interpretation:

$$g(x_1, x_2) = \theta_0 + \theta_1 x_1 + \theta_2 x_2$$

θ_0: distance from origin or bias

θ_1, θ_2: normal vector

(b). For template matching interpretation:

$$g(x_1, x_2) = \theta_0 + \theta_1 x_1 + \theta_2 x_2$$

θ_0: bias or threshold for strength of match

$\theta_1 x_1 + \theta_2 x_2$: Template

Q.4). Let $g(x_1, x_2) = 1 + 2x_1 + 3x_2$. Find the normal to the decision boundary defined by $g(x_1, x_2)$ and the distance of the decision boundary from the origin (note that for the distance computation the normal has to be normalized).

Soln:- Normal to the decision boundary defined by $g(x_1, x_2)$ } = (2, 3) Ans.

Distance of the decision boundary from the origin } = $\frac{1}{\sqrt{(2)^2+(3)^2}}$

$= \frac{1}{\sqrt{13}}$. Ans

2.5) When writing a discriminant function as $g(x) = \theta^T x$. where x and θ are vectors, explain where the bias coefficient in θ is and what is the required change needed to the feature vector x to allow writing the discriminant in this way.

Soln:- The bias coefficient in θ is at the beginning or at 1st position as shown below:

$$\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{bmatrix} \leftarrow \text{bias coefficient}$$

In feature vector x, we have to add 1 at the beginning or at 1st position as shown below:

$$x = \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \rightarrow \text{added 1 to the feature vector x.}$$

Q. 6). Write the expressions for the step and logistic (sigmoid) activation functions. Explain the advantage of the sigmoid activation over step activation. What happens to the sigmoid when θ is small?

Soln :→ Expression for step activation function :-

$$h_\theta(x) = \begin{cases} 1 & \text{if } \theta^T x > 0 \\ 0 & \text{otherwise.} \end{cases}$$

→ Expression for sigmoid activation function :-

$$h_\theta(x) = \frac{1}{1 + \exp(-\theta^T x)}$$

→ The advantage of the sigmoid activation over step activation is that we can also get intermediate values between 0 and 1 which can give more accurate ~~information~~ probability of an object belonging to a class.

→ ~~The value of sigmoid becomes approximately equal to 0.5 when θ is small.~~

→ Sigmoid activation function behaves like a step activation function when θ is small.

Q.7). Explain how the sigmoid is obtained using a linear function to model the log-likelihood ratio.

Soln:- Using a linear function to model the log-likelihood ratio:

$$\Rightarrow \log\left(\frac{P(y=1|x)}{P(y=0|x)}\right) = \theta^T x$$

On taking anti-log both sides, we get

$$\Rightarrow \frac{P(y=1|x)}{P(y=0|x)} = \exp(\theta^T x)$$

$$\Rightarrow P(y=1|x) = P(y=0|x)\exp(\theta^T x)$$

On replacing $P(y=0|x)$ with $(1-P(y=1|x))$, we get

$$\Rightarrow P(y=1|x) = (1-P(y=1|x))\exp(\theta^T x)$$

$$\Rightarrow P(y=1|x)(1+\exp(\theta^T x)) = \exp(\theta^T x)$$

$$\Rightarrow P(y=1|x) = \frac{\exp(\theta^T x)}{1+\exp(\theta^T x)}$$

$$\Rightarrow \boxed{P(y=1|x) = \frac{1}{1+\exp(-\theta^T x)} = h_\theta(x)}$$

→ Sigmoid activation.

Q.8). Write the derivative of a sigmoid and use it to compute the derivative of log-sigmoid.

Soln:- $\text{Sigmoid}(x) = \frac{1}{1+\exp(-x)}$

$$\frac{d}{dx}\,\text{sigmoid}(x) = \text{sigmoid}(x)\,(1-\text{sigmoid}(x))$$

$$= \frac{1}{1+\exp(-x)}\left\{1 - \frac{1}{1+\exp(-x)}\right\}$$

$$= \frac{1+\exp(-x)-1}{(1+\exp(-x))^2}$$

$$= \frac{\exp(-x)}{\{1+\exp(-x)\}^2}$$

$$\frac{d}{dx}\log\,\text{sigmoid}(x) = \frac{1}{\text{sigmoid}(x)} \cdot \text{sigmoid}(x) \cdot (1-\text{sigmoid}(x))$$

$$= 1 - \frac{1}{1+\exp(-x)}$$

$$= \frac{1+\exp(-x)-1}{1+\exp(-x)}$$

$$= \frac{\exp(-x)}{1+\exp(-x)}$$

Q.9). Explain how to compute the direction used for updating parameters in gradient descent. How is the size of the update controlled?

Soln:- We update the parameters in the opposite direction of the gradient of the objective function with respect to the parameters where the gradient gives the direction of the steepest ascent.

The size of the update is controlled by a hyperparameter called "learning rate".

Q.10). Explain the stop condition for gradient descent. Should the condition use the loss change or parameter change? Explain your answer.

Soln:- When the change in loss is less than a threshold in consecutive iteration, then we have to stop the process of gradient descent.

~~We should~~ Condition should use the loss change as we are minimizing the loss function and if loss doesn't change much then we can say that ~~we~~ it is converged but we don't know that smaller change in parameter will result in small change in loss. So, it should use loss change.

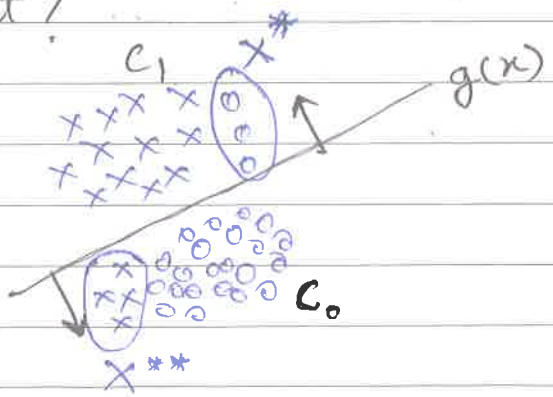
Q.11>. Explain what happens when using a learning rate that is too large or too small.

Soln :- • If learning rate is too large, it may fail to converge and overshoot the minimum.

• If learning rate is too small, it would take long time to converge and become computationally expensive.

Q.12>. Explain the empirical error loss is computed. What is the problem in using the empirical error loss in gradient descent?

Soln :-



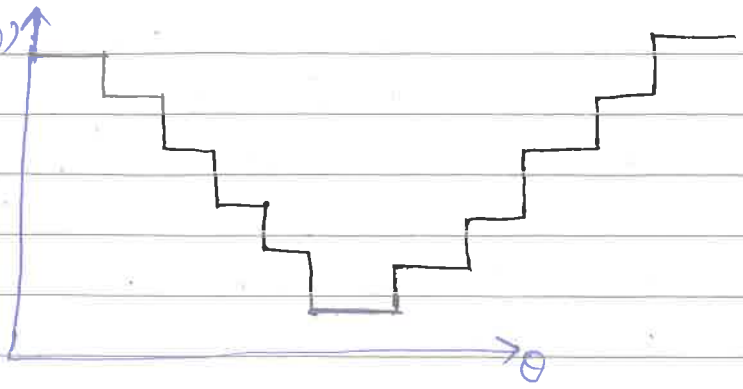
$$x^* = \{ x^{(i)} \mid \mathbb{1}(y^{(i)} = 0) \wedge \theta^T x > 0 \}$$

$$x^{**} = \{ x^{(i)} \mid \mathbb{1}(y^{(i)} = 1) \wedge \theta^T x < 0 \}$$

$E(\theta) = \#x^* + \#x^{**}$ = Total no. of errors.

$$\theta^* = \arg\min_{\theta} E(\theta)$$

need $\nabla E(\theta)$



The problem in using the empirical error loss in gradient descent is that there exist many values of 'θ' for same value of '$E(\theta)$'.

Q.13) Write the log likelihood for a binary classifier and show how to develop it into binary cross-entropy.

Sol:- $$l(\theta) = -\log\left(\prod_{x^{(i)} \in C_1} P(y=1|x^{(i)}) * \prod_{x^{(i)} \in C_0} P(y=0|x^{(i)})\right)$$

↳ negative log-likelihood.

$$= -\log\left(\prod_{i=1}^{m} P(y=1|x^{(i)})^{y^{(i)}} * P(y=0|x^{(i)})^{(1-y^{(i)})}\right)$$

$$= -\sum_{i=1}^{m} y^{(i)} \log(\underbrace{P(y=1|x^{(i)})}_{h_\theta(x)}) + (1-y^{(i)}) \log(1-\underbrace{P(y=1|x^{(i)})}_{h_\theta(x)})$$

$$\boxed{l(\theta) = -\sum_{i=1}^{m} y^{(i)} \log(h_\theta(x^{(i)})) + (1-y^{(i)}) \log(1-h_\theta(x^{(i)}))}$$

↳ binary cross-entropy.

Q.12> Write the gradient of the binary cross-entropy loss function with respect to the parameter vector when using a sigmoid activation function. Write the gradient descent update rule. when using this loss function. Explain the meaning of the update equation you wrote.

Sol:- binary cross-entropy loss function

$$l(\theta) = -\sum_{i=1}^{m} y^{(i)} \log(h_\theta(x^{(i)})) + (1-y^{(i)}) \log(1-h_\theta(x^{(i)}))$$

here,

assume $h_\theta(x)$ = sigmoid activation function.

then,

$$\frac{d\,l(\theta)}{d\theta} = -\sum_{i=1}^{m} y^{(i)} * \frac{1}{h_\theta(x^{(i)})} * h_\theta(x^{(i)})(1-h_\theta(x^{(i)}))\, x^{(i)}$$

$$+ \sum_{i=1}^{m} (1-y^{(i)}) * \frac{1}{(1-h_\theta(x^{(i)}))} * \left\{ -h_\theta(x^{(i)})(1-h_\theta(x^{(i)})) \right\} x^{(i)}$$

$$= -\sum_{i=1}^{m} y^{(i)}(1-h_\theta(x^{(i)}))\, x^{(i)} + (1-y^{(i)})(-h_\theta(x^{(i)}))\, x^{(i)}$$

$$= -\sum_{i=1}^{m} \left\{ y^{(i)} - y^{(i)} h_\theta(x^{(i)}) - h_\theta(x^{(i)}) + y^{(i)} h_\theta(x^{(i)}) \right\} x^{(i)}$$

gradient of binary cross-entropy

$$\boxed{\frac{d}{d\theta}(-l(\theta)) = \sum_{i=1}^{m} \{ h_\theta(x^{(i)}) - y^{(i)} \}\, x^{(i)}}$$

Gradient descent update rule :

$$\theta_{new} = \theta_{old} - \eta \underbrace{\left\{ \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})\, x^{(i)} \right\}}_{\frac{d}{d\theta}(-l(\theta))}$$

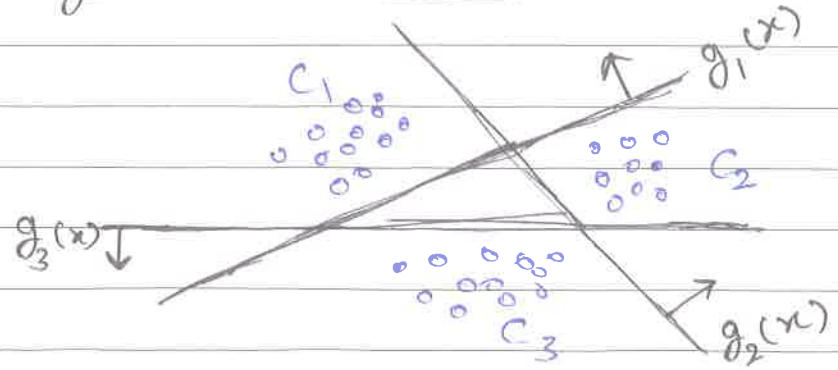
learning rate.

Here, we calculate the gradient in order to get the direction so that we can update parameters in this direction.

learning rate decides the size of the update and then subtract them from old parameters to get new parameters.

Q.15). Explain the two strategies for multi-class classification (one against all others, one against each other). In which strategy it is it easier to discriminate the data?

Soln :- • One against all others



Here,

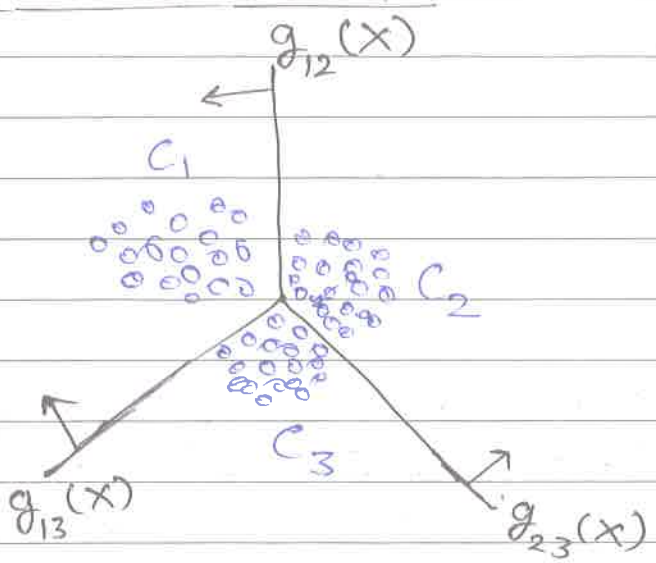
we have 'k' discriminant functions for 'k' classes.

$g_1(x), \ldots, g_K(x)$

$$\hat{y} = \underset{j}{\text{argmax}}\ g_j(x)$$

Here, we choose a discriminant function for which data has maximum value.

- One against each other



Here, we have $\frac{K(K-1)}{2}$ discriminant functions for 'k' classes.

$$\hat{y} = \underset{i}{\text{argmax}} \sum_{j \neq i} g_{ij}(x)$$

In

- One against each other strategy, it is easier to discriminate the data.

Q.16) Explain the template matching interpretation of linear discriminants. Using this interpretation what is the meaning of using multiple linear discriminant functions.

Solu :- Template Matching Interpretation

→ Rows of θ^T (columns of θ) are templates.

→ With K rows of θ^T we have K templates (one template per class).

→ $\theta^T x$ measures how well x matches each of the K templates (dot product of x with rows of θ^T).

→ High similarity to a template of a particular class indicates high membership in this class.

The meaning of using multiple linear discriminant functions is to have multiple templates for multi-class classification problem.

Q.17) Explain the need for using softmax in the output layer of a multi-class classifier.

Solu :- We know that, logistic regression produces a decimal between 0 and 1. For example,

a logistic regression output of 0.8 from an email classifier suggests an 80% chance of an email being spam and a 20% chance of it being not spam. Clearly, the sum of the probabilities of an email being either spam or not spam is 1.

Softmax extends this idea into a multi-class world. That is, softmax assigns decimal probabilities to each class in a multi-class problem. Those decimal probabilities must add upto 1.

Due to this, softmax is needed in the output layer of a multi-class classifier.

Q.18). Write the derivative of softmax and use it to compute the derivative of log-softmax.

Soln :- Given :-

$$\text{Softmax}(z_i) = \frac{\exp(z_i)}{\sum_{i=1}^{k} \exp(z_i)}$$

$$\frac{\partial}{\partial z_i} \text{Softmax}(z_i)$$

$$= \text{softmax}(z_i)(\delta_{ij} - \text{softmax}(z_i))$$

$$\frac{\partial}{\partial z_i}(\log\text{-softmax}(z_i)) = \frac{1}{\text{softmax}(z_i)} * \{\text{softmax}(z_i)(\delta_{ij} - \text{softmax}(z_i))$$

$$= (\delta_{ij} - \text{softmax}(z_i))$$

Q.19) Write the log-likelihood for a multi-class classifier and show how to develop it into categorical cross-entropy.

Soln:- $$l(\theta) = -\log\left(\prod_{i=1}^{m}\prod_{j=1}^{k} \underbrace{P(y^{(i)}=j \mid x^{(i)})}_{h_{\theta_j}(x^{(i)})}^{1(y^{(i)}=j)}\right)$$

↳ negative log-likelihood

$$= -\sum_{i=1}^{m}\sum_{j=1}^{k} 1(y^{(i)}=j)\log(P(y^{(i)}=j \mid x^{(i)}))$$

$$\boxed{l(\theta) = -\sum_{i=1}^{m}\sum_{j=1}^{k} 1(y^{(i)}=j)\log(h_{\theta_j}(x^{(i)}))}$$

↳ Categorical cross-entropy

Q.20) Write the gradient of the categorical cross-entropy loss function with respect to the parameter vector when using a softmax activation function. Write the gradient descent update rule when using this loss function. Explain the meaning of

the update equation you wrote.

Soln:- $l(\theta) = -\sum_{i=1}^{m}\sum_{j=1}^{k} 1(y^{(i)}=j)\log\left(h_{\theta_j}(x^{(i)})\right)$

↳ Categorical cross-entropy loss fⁿ.

Here,

assume $h_{\theta_j}(x^{(i)})$ = softmax activation function

Then,

$$\frac{\partial}{\partial\theta_j}(l(\theta)) = -\sum_{i=1}^{m} 1(y^{(i)}=j) * \frac{1}{h_{\theta_j}(x^{(i)})}\{h_{\theta_j}(x^{(i)}) * \left(1(y^{(i)}=j) - h_{\theta_j}(x^{(i)})\right)x^{(i)}$$

$$\boxed{\frac{\partial}{\partial\theta_j}(l(\theta)) = \sum_{i=1}^{m}\left(h_{\theta_j}(x^{(i)}) - 1(y^{(i)}=j)\right)x^{(i)}}$$

↳ gradient of categorical cross-entropy

Gradient descent update rule:

$$\theta_j^{(new)} = \theta_j^{(old)} - \eta\sum_{i=1}^{m}\left(h_{\theta_j}(x^{(i)}) - 1(y^{(i)}=j)\right)x^{(i)}$$

η : learning rate; $\sum_{i=1}^{m}\left(h_{\theta_j}(x^{(i)}) - 1(y^{(i)}=j)\right)x^{(i)} = \frac{\partial}{\partial\theta_j}(l(\theta))$

Here, we calculate the gradient in order to get the direction so that we can update parameter in this direction.

Learning rate decides the size of the update and then subtract them from old parameter to get new parameter.

NEURAL NETWORKS

Q.1) Given a two layer network where the number of hidden units is larger than the number of inputs explain the dimensionality increase interpretation of the hidden layer. Explain what is the possible benefit of such dimensionality increase.

Soln:- Dimensionality increase interpretation of the hidden layer:

no. of hidden units > no. of inputs.

2D → 5D

$(X_1, X_2) \longrightarrow (X_1, X_2, X_1^2, X_2^2, X_1 \cdot X_2)$

input → hidden units

The possible benefit of such dimensionality increase is to establish relationship in higher dimensional space.

Q.2) Given a two layer network where the no. of hidden units is smaller than the no. of inputs explain the dimensionality decrease interpretation of the hidden layer. Explain what is the possible benefit of dimensionality decrease.

Soln :- Dimensionality reduction interpretation of the hidden layer:

no. of hidden units < no. of inputs.

6D → 4D

$(x_1, x_2, x_3, x_4, x_5, x_6) \rightarrow (x_1, x_3, x_4, x_6)$

input → hidden units.

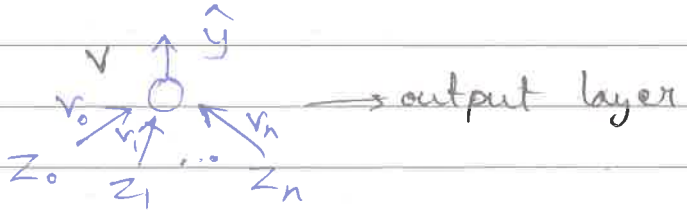
The possible benefit of dimensionality decrease is to extract important features and discard the features which are correlated with each other.

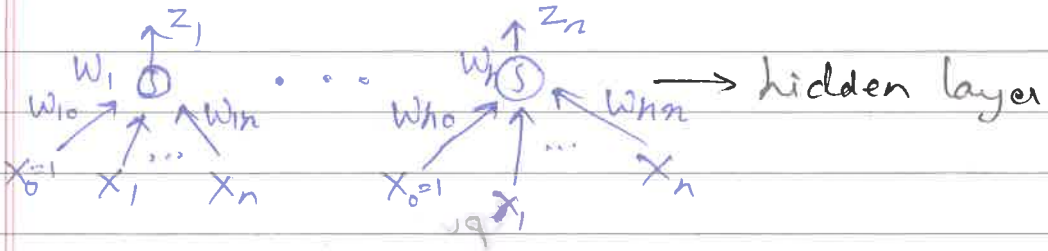
Q.3) Given the update equations for the output layer in a 2 layer network, explain why we cannot use directly the same equations for the hidden layer. Assume that both layers have the same activation function.

Soln :- We cannot use directly the same equations for the hidden layer because it will not take gradient of loss with respect to the weight parameters of hidden layer instead it will take gradient of loss with respect to the weight parameters of the output layer.

Q.4) Apply the chain rule to compute the gradients with respect to both hidden layer and output layer parameters of a 2-layer regression network with a single output where the hidden layers use sigmoid activation and the output layer uses linear activation.

Soln:-





Loss: $E = \frac{1}{2} \sum_{i=1}^{m} (\hat{y}^{(i)} - y^{(i)})^2$ ← $h_\theta(x^{(i)})$

Outputs: $\hat{y} = V^T z = V_0 + V_1 z_1 + \cdots + V_n z_n$

$z_j = \text{sigmoid}(w_j^T x)$

Chain rule for output layer:

$$\frac{\partial E}{\partial V} = \frac{\partial E}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial V} = \frac{1}{2} \times 2 \sum_{i=1}^{m} (\hat{y}^{(i)} - y^{(i)}) z^{(i)}$$

$$= \sum_{i=1}^{m} (\hat{y}^{(i)} - y^{(i)}) z^{(i)}$$

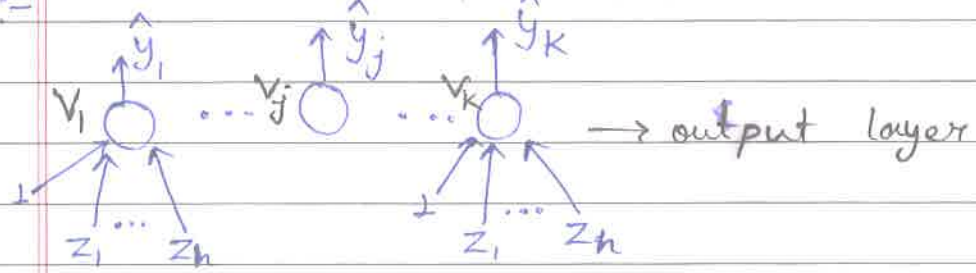
Chain rule for hidden layer:

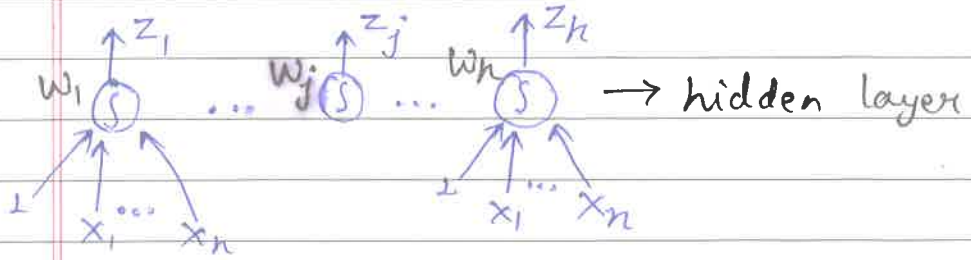
$$\frac{\partial E}{\partial w_j} = \frac{\partial E}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial z_j} \frac{\partial z_j}{\partial w_j}$$

$$\frac{\partial E}{\partial w_j} = \sum_{i=1}^{m} (\hat{y}^{(i)} - y^{(i)}) V_j z_j^{(i)} (1 - z_j^{(i)}) x^{(i)}$$

Q.5). Apply the chain rule to compute the gradients with respect to both hidden layer and output layer parameters of a 2-layer regression network with multiple outputs where the hidden layers use sigmoid activation and the output layer uses linear activation

Soln :-





Loss :- $$E = \frac{1}{2} \sum_{i=1}^{m} \sum_{l=1}^{k} (\hat{y}_l^{(i)} - y_l^{(i)})^2$$

Outputs : $$\hat{y}_j = V_j^T z = V_{j0} + V_{j1} z_1 + \cdots + V_{jh} z_h$$

$$z_j = \text{sigmoid}(w_j^T x)$$

Chain rule for output layer :

$$\frac{\partial E}{\partial V_j} = \frac{\partial E}{\partial \hat{y}_j} \frac{\partial \hat{y}_j}{\partial V_j} = \frac{1}{\cancel{2}} \times \cancel{2} \sum_{i=1}^{m} (\hat{y}_j^{(i)} - y_j^{(i)}) z^{(i)}$$

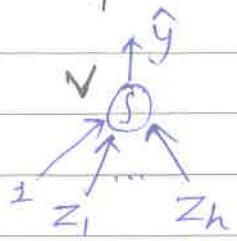
Chain rule for hidden layer :

$$\frac{\partial E}{\partial w_j} = \sum_{\ell=1}^{k} \frac{\partial E}{\partial \hat{y}_\ell} \frac{\partial \hat{y}_\ell}{\partial z_j} \frac{\partial z_j}{\partial w_j}$$

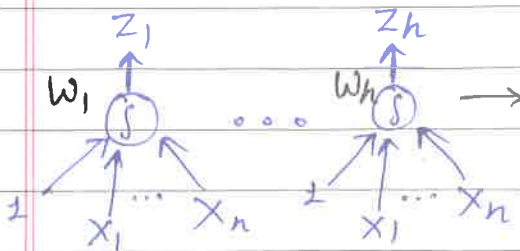
$$= \sum_{i=1}^{m} \sum_{\ell=1}^{k} (\hat{y}_\ell^{(i)} - \hat{y}_\ell^{(i)}) V_{\ell j} z_j^{(i)} (1 - z_j^{(i)}) x^{(i)}$$

Q.6). Apply the chain rule to compute the gradients with respect to both hidden layer and output layer parameters of a 2-layer binary classification network where the hidden layers use sigmoid activation and the output layer uses sigmoid activation.

Soln :-



→ output layer



→ hidden layer

Loss :

$$L(\theta) = \prod_{i=1}^{m} P(y=1|x^{(i)})^{y^{(i)}} \underbrace{(1 - P(y=1|x^{(i)}))}_{P(y=0|x^{(i)})}^{1-y^{(i)}}$$

$$l(\theta) = -\log(L(\theta))$$

$$= -\sum_{i=1}^{m} y^{(i)} \log(\underbrace{P(y=1|x^{(i)})}_{\hat{y}^{(i)}}) + (1-y^{(i)}) \log(1-\underbrace{P(y=1|x^{(i)})}_{\hat{y}^{(i)}})$$

$$\boxed{l(\theta) = -\sum_{i=1}^{m} y^{(i)} \log(\hat{y}^{(i)}) + (1-y^{(i)}) \log(1-\hat{y}^{(i)})}$$

~~Chain rule for~~ θ

Outputs:

$$\hat{y} = \text{sigmoid}(v^T z)$$

$$z_j = \text{sigmoid}(w_j^T x)$$

Chain rule for output layer:

$$\frac{\partial l(\theta)}{\partial v} = \frac{\partial l}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial v}$$

$$= \sum_{i=1}^{m} \left(y^{(i)} \frac{1}{\hat{y}^{(i)}} - (1-y^{(i)}) \frac{1}{1-\hat{y}^{(i)}} \right) \hat{y}^{(i)} (1-\hat{y}^{(i)}) z^{(i)}$$

$$= \sum_{i=1}^{m} (\hat{y}^{(i)} - y^{(i)}) z^{(i)}$$

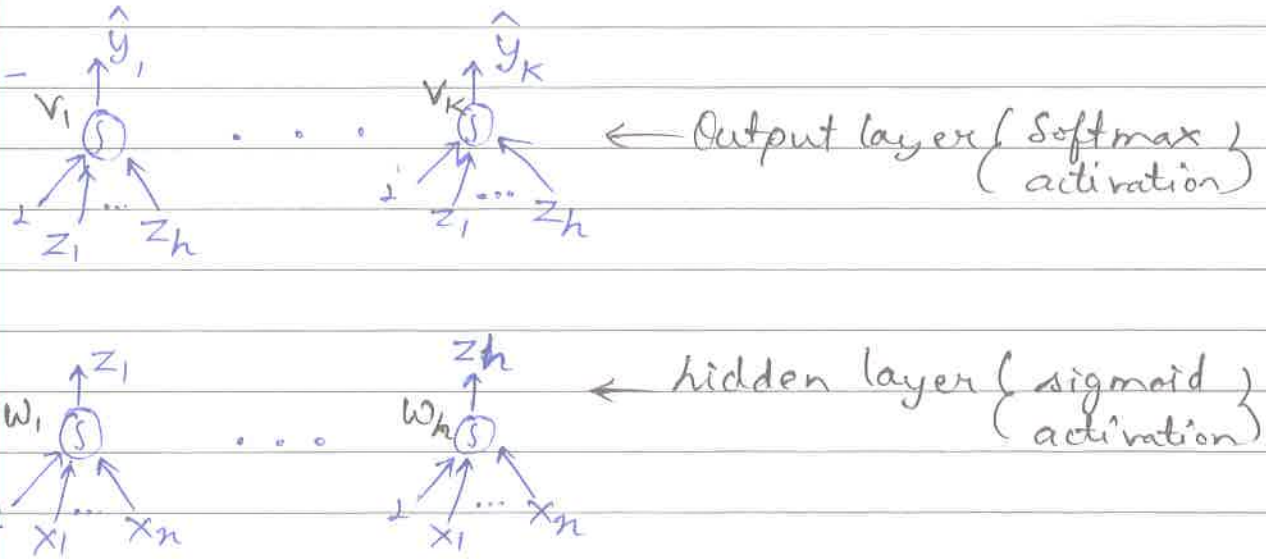
Chain rule for hidden layer :-

$$\frac{\partial l(\theta)}{\partial w_j} = \frac{\partial l}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial z_j} \frac{\partial z_j}{\partial w_j}$$

$$\frac{\partial l}{\partial w_j} = \sum_{i=1}^{m} (\hat{y}^{(i)} - y^{(i)})\, v_j\, z_j^{(i)} (1 - z_j^{(i)})\, x^{(i)}$$

Q.7) Apply the chain rule to compute the gradients with respect to both hidden layer and output layer parameters of a 2-layer multiclass classification network where the hidden layers use sigmoid activation and the output layer uses softmax activation.

Soln :-



Loss :-

$$L(\theta) : \prod_{i=1}^{m} \prod_{j=1}^{k} P(y=j \mid x^{(i)})^{\mathbb{1}(y^{(i)}=j)}$$

$$l(\theta) = -\log(L(\theta))$$

$$= -\sum_{i=1}^{m} \sum_{j=1}^{k} \mathbb{1}(y^{(i)}=j) \log\left(\underbrace{P(y=j \mid x^{(i)})}_{\hat{y}_j^{(i)}}\right)$$

$$\boxed{l(\theta) = -\sum_{i=1}^{m} \sum_{j=1}^{k} \mathbb{1}(y^{(i)}=j) \log(\hat{y}_j^{(i)})}$$

Outputs:-

$$\hat{y}_j = \text{softmax}(v_j^T z)$$

$$z_j = \text{sigmoid}(w_j^T x)$$

Chain rule for output layer:

$$\frac{\partial l}{\partial v_j} = \frac{\partial l}{\partial y_j} \frac{\partial y_j}{\partial v_j}$$

$$= -\sum_{i=1}^{m} (\hat{y}_j^{(i)} - y_j^{(i)}) z^{(i)}$$

Chain rule for hidden layer:

$$\frac{\partial l}{\partial w_j} = \sum_{l=1}^{k} \frac{\partial l}{\partial \hat{y}_l} \frac{\partial \hat{y}_l}{\partial z_j} \frac{\partial z_j}{\partial w_j}$$

$$= \sum_{i=1}^{m} \left(\sum_{l=1}^{k} (\hat{y}_l^{(i)} - y_l^{(i)}) v_{lj} \right) z_j^{(i)} (1 - z_j^{(i)}) x^{(i)}$$

Q.8) Describe how weights in the network should be initialized and explain why they are initialized in this way.

Soln:- Weights in the layer of a network should be initialized randomly from a normal distribution with mean $\mu = 0$ and

variance $\sigma^2 = \frac{1}{\text{no. of neurons in the previous layer}}$.

Ensuring zero-mean and maintaining the value of the variance of the input of every layer guarantees no exploding / vanishing gradient. Due to this reason, they are initialized in this way.

COMPUTATION GRAPHS

Q.1) Explain the advantage of using computation graphs. Describe what is computed during the forward pass and backward pass of the network during backpropagation. Explain what each node in the graph needs to be able to compute and what information it needs to store (and why).

Solu:- Advantage of using computation graphs is its efficiency in calculating output node values and also computing gradients for a node individually. Due to this property we can do multiple operations in parallel.

All the intermediate node values is calculated in the forward pass. ~~and~~ gradient gradients is computed and pushes towards beginning from the end of the network in backward pass of the network during backpropagation.

Each node in the graph needs to be able to compute the gradient of the output values with respect to each input to the node and it needs to store output value of each individual node which is computed in forward pass because this intermediate output value is used for computing gradients in backward pass.

Q.2> Given a two layer network where the hidden layer output is given by $z = f_1(W_1, x)$ and the output layer output is given by $\hat{y} = f_2(W_2, z)$ express the output $\hat{y}$ as a function composition between $f_1()$ and $f_2()$. Assuming the output layer has a single node, express the L_2 loss for a batch $\{x^{(i)}, y^{(i)}\}_{i=1}^{m}$ in terms of the composite function you wrote and use the chain rule to write its derivatives with respect to W_1 and W_2. You do not need to compute the actual derivatives of $f_1()$ and $f_2()$ as they are not specified. Repeat the question with cross entropy loss instead of L_2.

Sol^n:-

$$\hat{y} = f_2(W_2, f_1(W_1, x))$$

L_2 loss for batch $\{x^{(i)}, y^{(i)}\}_{i=1}^{m}$:

$$\text{loss} = \sum_{i=1}^{m} \left(\hat{y}^{(i)} - y^{(i)}\right)^2$$

$$L(\text{loss}) = \sum_{i=1}^{m} \left(f_2(W_2, f_1(W_1, x^{(i)})) - y^{(i)} \right)^2$$

$$\frac{\partial L}{\partial W_1} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial f_1}$$

$$\frac{\partial L}{\partial W_1} = \frac{\partial L}{\partial f_2} \frac{\partial f_2}{\partial f_1} \frac{\partial f_1}{\partial W_1}$$

$$\frac{\partial L}{\partial W_2} = \frac{\partial L}{\partial f_2} \frac{\partial f_2}{\partial W_2}$$

Here,

$$\frac{\partial L}{\partial f_2} = 2(\hat{y} - y)$$

where,

$$\hat{y} = f_2(W_2, f_1(W_1, x))$$

With cross-entropy loss instead of L_2 :

$$L(\text{loss}) = -\sum_{i=1}^{m} y^{(i)} \log\left(f_2(W_2, f_1(W_1, x^{(i)}))\right) + (1 - y^{(i)}) \log\left(1 - f_2(W_2, f_1(W_1, x^{(i)}))\right)$$

$$\frac{\partial L}{\partial W_1} = \frac{\partial L}{\partial f_2} \frac{\partial f_2}{\partial f_1} \frac{\partial f_1}{\partial W_1} \; ; \; \frac{\partial L}{\partial W_2} = \frac{\partial L}{\partial f_2} \frac{\partial f_2}{\partial W_2}$$

Here,
$$\frac{\partial L}{\partial f_2} = -\sum_{i=1}^{m} \frac{y^{(i)}}{f_2(W_2, f_1(W_1, x^{(i)}))} + \frac{(1-y^{(i)})}{1-f_2(W_2, f_1(W_1, x^{(i)}))}$$

Q.3). Repeat the previous question for a multiple output network.

Soln :-

$$\hat{y}_j = f_2(W_2, f_1(W_1, X))$$

L_2 loss for batch $\{x^{(i)}, y^{(i)}\}_{i=1}^{m}$:

$$L(\text{loss}) = \sum_{i=1}^{m} \sum_{l=1}^{k} (\hat{y}_l^{(i)} - y_l^{(i)})^2$$

$$\text{loss} = \sum_{i=1}^{m} \sum_{l=1}^{k} (f_{2_l}(W_2, f_1(W_1, x^{(i)})) - y_l^{(i)})^2$$

$$\frac{\partial L}{\partial W_{1j}} = \sum_{l=1}^{k} \frac{\partial L}{\partial f_{2l}} \frac{\partial f_{2l}}{\partial f_{1j}} \frac{\partial f_{1j}}{\partial W_{1j}}$$

$$\frac{\partial L}{\partial W_{2j}} = \frac{\partial L}{\partial f_{2j}} \frac{\partial f_{2j}}{\partial W_{2j}}$$

Here,
$$\frac{\partial L}{\partial f_{2j}} = 2 \sum_{i=1}^{m} (f_{2_j}(W_2, f_1(W_1, x^{(i)})) - y_j^{(i)})$$

With cross-entropy loss instead of L_2 :-

$$L(\text{loss}) = -\sum_{i=1}^{m}\sum_{j=1}^{k} \mathbb{1}(y^{(i)}=j) \log\left(\hat{y}_j^{(i)}\right)$$

$$= -\sum_{i=1}^{m}\sum_{j=1}^{k} \mathbb{1}(y^{(i)}=j) \log\left(f_{2j}(W_2, f_1(W_1, x^{(i)}))\right)$$

$$\frac{\partial L}{\partial W_{1j}} = \sum_{l=1}^{k} \frac{\partial L}{\partial f_{2l}} \frac{\partial f_{2l}}{\partial f_{1j}} \frac{\partial f_{1j}}{\partial W_{1j}}$$

$$\frac{\partial L}{\partial W_{2j}} = \frac{\partial L}{\partial f_{2j}} \frac{\partial f_{2j}}{\partial W_{2j}}$$

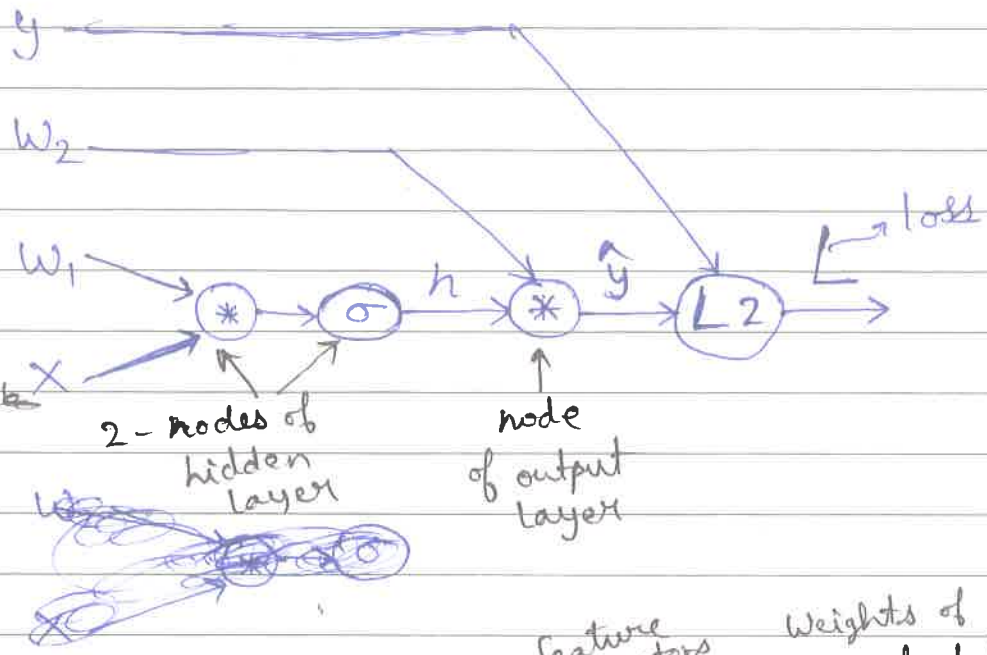
Here,

$$\frac{\partial L}{\partial f_{2j}} = -\sum_{i=1}^{m} \frac{\mathbb{1}(y^{(i)}=j)}{f_{2j}(W_2, f_1(W_1, x^{(i)}))}$$

2.4) Write the computation graph for a two layer regression network with 2 nodes in the hidden layer and 1 node in the output layer. Assume sigmoid activation in the hidden layer and linear activation in the output layer. Assume the feature vectors are 3-dimensional. List all the inputs of the computation graphs and write all the derivatives needed to compute the loss gradient with respect to

network parameters when using the backpropagation pass.

Soln :- Computation graph:



Inputs of the computation graph } = X, W_1, W_2, y.

X : feature vectors
W_2 : weights of output layer
W_1 : weights of hidden layer
y : actual output

All gradients needed to compute are :

$$L = (\hat{y} - y)^2$$
$$h = \sigma(XW_1)$$
$$\hat{y} = hW_2$$

$$\frac{\partial L}{\partial \hat{y}} = 2(\hat{y} - y) \qquad \frac{\partial \hat{y}}{\partial W_2} = h$$

$$\frac{\partial \hat{y}}{\partial h} = W_2 \qquad \frac{\partial h}{\partial W_1} = h(1-h)X$$

$$\frac{\partial L}{\partial W_2} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial W_2} = 2(\hat{y}-y)\cdot h$$

$$= \underline{2(\hat{y}-y)\cdot \sigma(XW_1)}$$

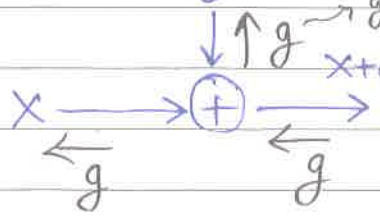
$$\frac{\partial L}{\partial h} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial h} = 2(\hat{y}-y)\cdot W_2$$

$$\frac{\partial L}{\partial W_1} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial h} \frac{\partial h}{\partial W_1} = 2(\hat{y}-y)\cdot W_2 \cdot h(1-h)\, X$$

$$= 2(\hat{y}-y)\cdot W_2 \cdot \sigma(XW_1)(1-\sigma(XW_1))\, X$$

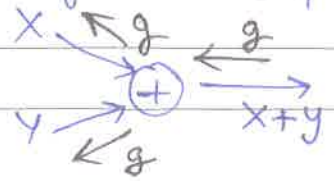
Q.5) Describe the back flow patterns for the following nodes: addition of a constant, addition of two inputs, multiplication by a constant, multiplication of two inputs, max of two inputs, min of two inputs.

Soln:- • addition of a constant :-



$$\frac{d}{dx}(x+c) = 1$$

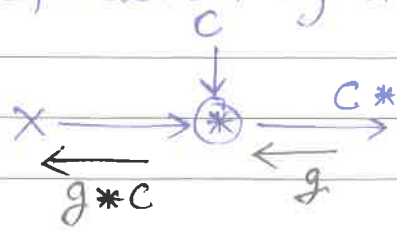
• addition of two inputs :-



$$\frac{\partial}{\partial x}(x+y) = 1$$

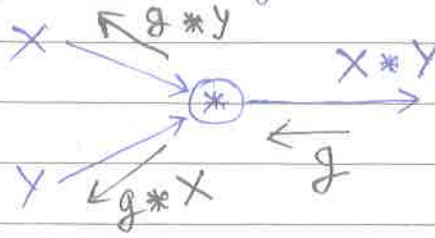
$$\frac{\partial}{\partial y}(x+y) = 1$$

• Multiplication by a constant:



$$\frac{\partial}{\partial x}(c * x) = c$$

• Multiplication of two inputs:



$$\frac{\partial}{\partial x}(x * y) = y$$

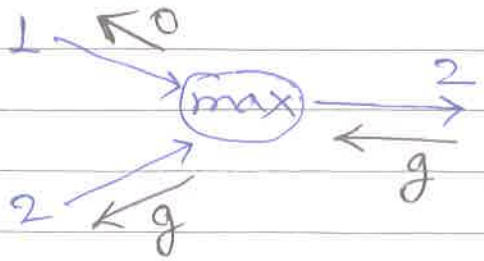
$$\frac{\partial}{\partial y}(x * y) = x$$

• Max of two inputs:

$$\max(x, y) = \begin{cases} x & \text{if } x \geq y \\ y & \text{otherwise} \end{cases}$$

$$\frac{\partial}{\partial x} \max(x, y) = \begin{cases} 1 & \text{if } x \geq y \\ 0 & \text{otherwise} \end{cases}$$

$$\frac{\partial}{\partial y} \max(x, y) = \begin{cases} 0 & \text{if } x \geq y \\ 1 & \text{otherwise} \end{cases}$$

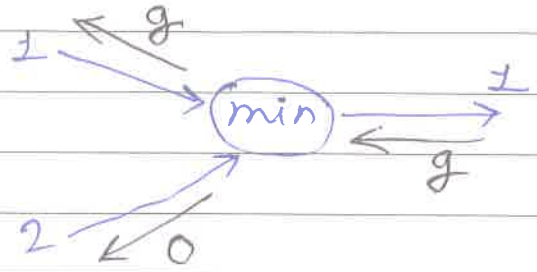


• Min of two inputs:

$$\min(x, y) = \begin{cases} x & \text{if } x \leq y \\ y & \text{otherwise} \end{cases}$$

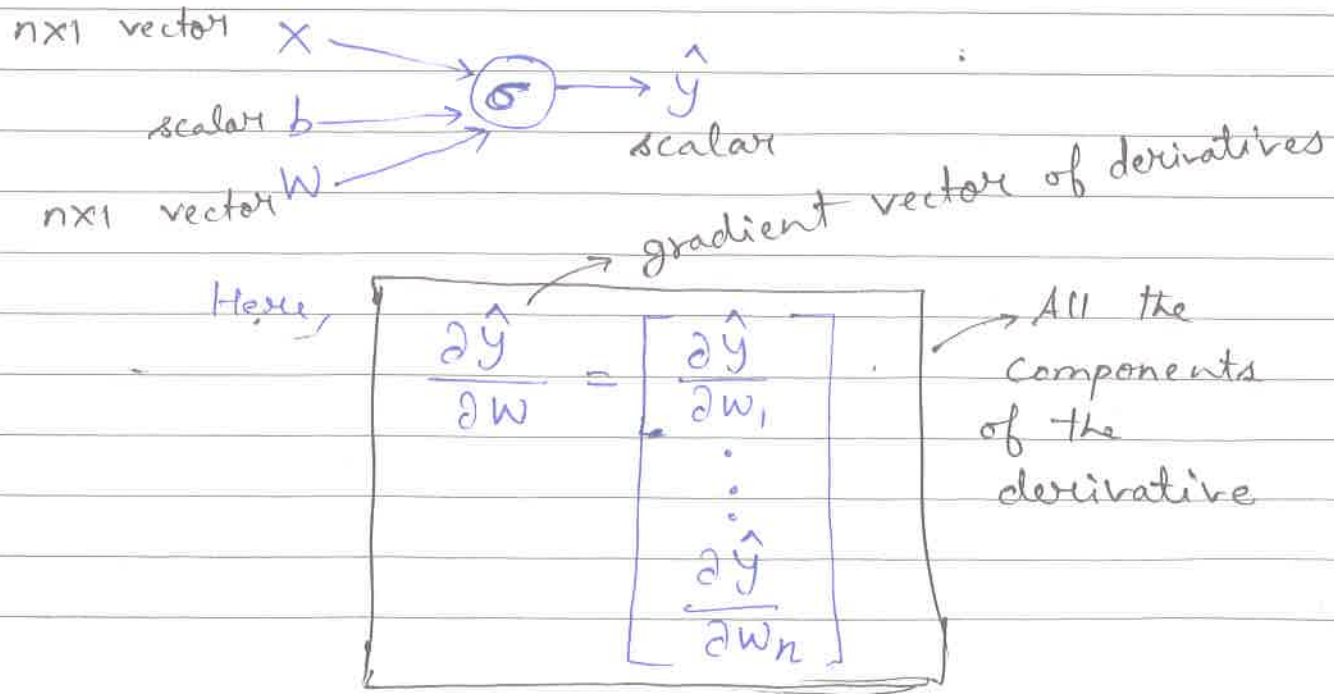
$$\frac{\partial}{\partial x} \min(x,y) = \begin{cases} 1 & \text{if } x \leq y \\ 0 & \text{otherwise} \end{cases}$$

$$\frac{\partial}{\partial y} \min(x,y) = \begin{cases} 0 & \text{if } x \leq y \\ 1 & \text{otherwise} \end{cases}$$



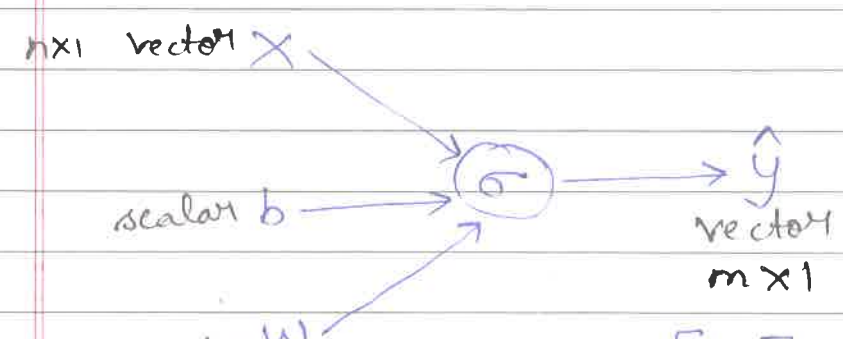
Q.6) Explain what derivative you expect when the input to a node is a vector and the output is a scalar. Write all the components of the derivative.

Soln:- When the input to a node is a vector and the output is a scalar then we will get a rank 1 tensor or gradient vector.



Q.7). Explain what derivative you expect when the input to a node is a vector and the output is a vector. Write all the components of the derivative.

Soln :- When the input to a node is a vector and the output is a vector then ~~we~~ we will get rank 2 tensor of derivatives.



$$W = \begin{bmatrix} w_1 \\ \vdots \\ w_n \end{bmatrix} \qquad \hat{y} = \begin{bmatrix} \hat{y}_1 \\ \vdots \\ \hat{y}_m \end{bmatrix}$$

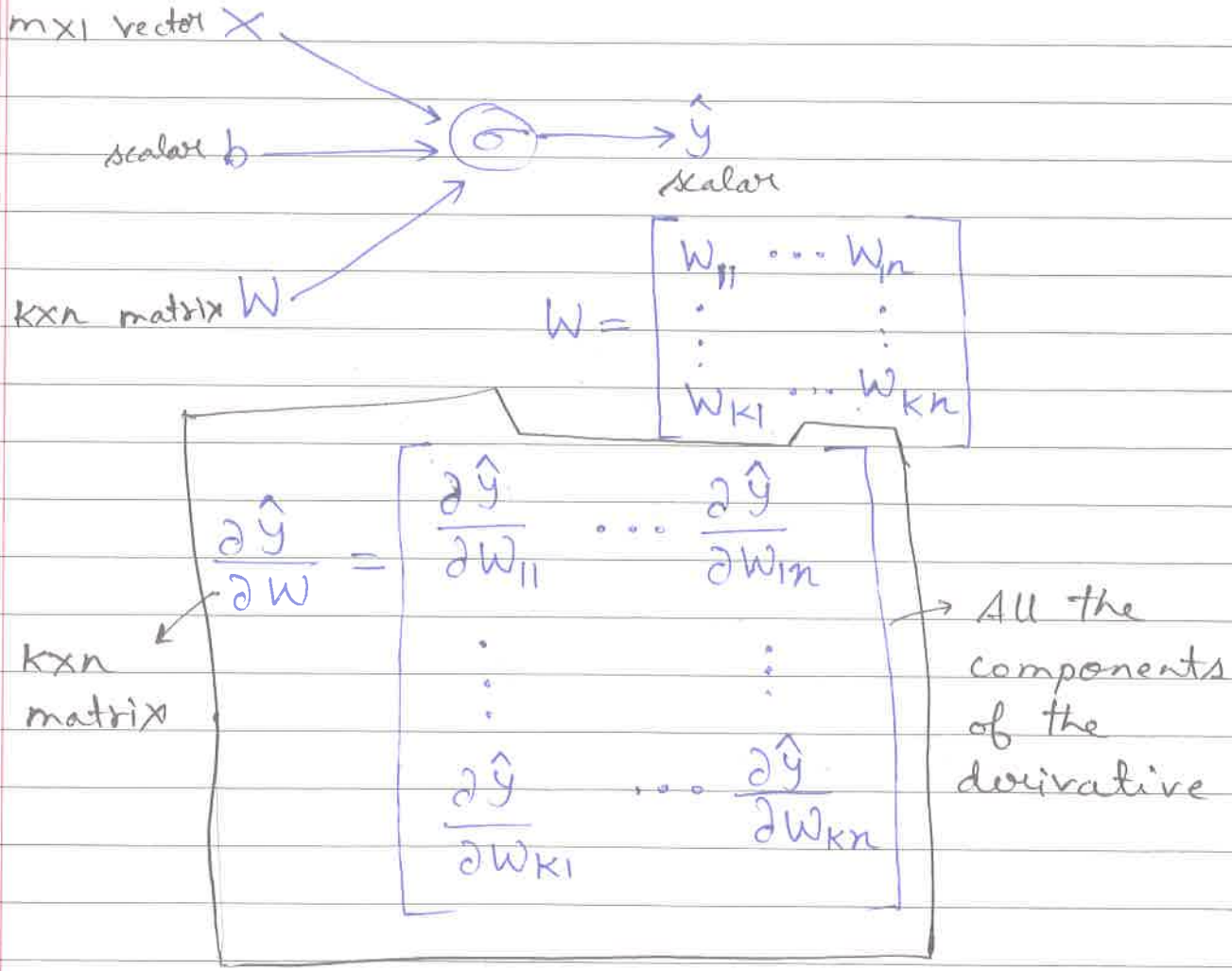
$$\frac{\partial \hat{y}}{\partial w} = \begin{bmatrix} \frac{\partial \hat{y}_1}{\partial w_1} & \cdots & \frac{\partial \hat{y}_1}{\partial w_n} \\ \vdots & & \\ \frac{\partial \hat{y}_m}{\partial w_1} & \cdots & \frac{\partial \hat{y}_m}{\partial w_n} \end{bmatrix}_{(m \times n)}$$

→ All the components of the derivative.

↓ It is a rank 2 tensor. Shape of this matrix is $m \times n$.

Q.8). Explain what derivative you expect when the input to a node is a matrix and the output is a scalar. Write all the components of the derivative.

Solu:- When the input to a node is a matrix and the output is a scalar then we will get rank 2 tensor of derivatives.



Q.9). Write the chain rule for the composition of vector valued vector functions F and G. Pay attention to the order of the Jacobian matrices you write.

Solu:- → Let $F: R^k \to R^m$ and $G: R^h \to R^k$ be vector valued functions.

→ The composition of F and G is the function:

$F \circ G : R^h \to R^m$ given by:

$$(F \circ G)(P) = F(G(P))$$

$m \times 1$ ← F, $k \times 1$ ← $G(P)$, $n \times 1$ ← P

Chain rule for the composition of vector valued functions:

Product of matrices

$$\underset{m \times n}{D(F \circ G)(P)} = \underbrace{\underset{m \times k}{D\underline{F}(G(P))}}_{\text{Compute DF then assign } G(P) \text{ instead of } P} * \underbrace{\underset{k \times n}{D\underline{G}(P)}}$$

$$m \times k \quad * \quad k \times n \quad = \quad m \times n$$

rank 2 tensor | rank 2 tensor | rank 2 tensor

Example:-

$$F(x,y,z) = \begin{bmatrix} 2xy \\ y - z \end{bmatrix} \qquad G(x,y) = \begin{bmatrix} x - 4y \\ xy \\ x + y \end{bmatrix}$$

$$DF(x,y,z) = \begin{bmatrix} \frac{\partial}{\partial x}(2xy) & \frac{\partial}{\partial y}(2xy) & \frac{\partial}{\partial z}(2xy) \\ \frac{\partial}{\partial x}(y-z) & \frac{\partial}{\partial y}(y-z) & \frac{\partial}{\partial z}(y-z) \end{bmatrix} = \begin{bmatrix} 2y & 2x & 0 \\ 0 & 1 & -1 \end{bmatrix}$$

Using the chain rule

→ first compute DF and then plug in G(x,y)

$$DF(G(x,y)) = \begin{bmatrix} 2(xy) & 2(x-4y) & 0 \\ 0 & 1 & -1 \end{bmatrix}$$

$$DG(x,y) = \begin{bmatrix} \frac{\partial}{\partial x}(x-4y) & \frac{\partial}{\partial y}(x-4y) \\ \frac{\partial}{\partial x}(xy) & \frac{\partial}{\partial y}(xy) \\ \frac{\partial}{\partial x}(x+y) & \frac{\partial}{\partial y}(x+y) \end{bmatrix} = \begin{bmatrix} 1 & -4 \\ y & x \\ 1 & 1 \end{bmatrix}$$

Using the chain rule:

$$D(F \circ G)(x,y) = DF(G(x,y)) * DG(x,y)$$

$$= \underset{2\times 3}{\begin{bmatrix} 2xy & 2x-8y & 0 \\ 0 & 1 & -1 \end{bmatrix}} \underset{3\times 2}{\begin{bmatrix} 1 & -4 \\ y & x \\ 1 & 1 \end{bmatrix}}$$

$$= \begin{bmatrix} 2xy + 2xy - 8y^2 & -8xy + 2x^2 - 8xy \\ y-1 & x-1 \end{bmatrix}$$

$$= \underset{2\times 2}{\begin{bmatrix} 4xy - 8y^2 & 2x^2 - 16xy \\ y-1 & x-1 \end{bmatrix}}$$

Q.10) Let $F(x,y,z) = [3xy \quad y-z]^T$ and $G(x,y) = [x-5y \quad xy \quad x-y]^T$ write the composition $(f \circ G)(x,y)$. Compute the Jacobian matrix of the composition in two ways: directly and by using the chain rule. Make sure the result you compute in both ways is identical.

Soln:-

$$F(x,y,z) = \begin{bmatrix} 3xy \\ y-z \end{bmatrix} \qquad G(x,y) = \begin{bmatrix} x-5y \\ xy \\ x-y \end{bmatrix}$$

$$(F \circ G)(x,y) = \begin{bmatrix} 3(x-5y)(xy) \\ (xy)-(x-y) \end{bmatrix} = \begin{bmatrix} 3x^2y - 15xy^2 \\ xy - x + y \end{bmatrix}$$

Computing the Jacobian matrix directly:

$$D(F \circ G)(x,y) = \begin{bmatrix} \frac{\partial}{\partial x}(3x^2y - 15xy^2) & \frac{\partial}{\partial y}(3x^2y - 15xy^2) \\ \frac{\partial}{\partial x}(xy - x + y) & \frac{\partial}{\partial y}(xy - x + y) \end{bmatrix}$$

$$D(F \circ G)(x,y) = \begin{bmatrix} 6xy - 15y^2 & 3x^2 - 30xy \\ y - 1 + 0 & x - 0 + 1 \end{bmatrix}$$

$$= \begin{bmatrix} 6xy - 15y^2 & 3x^2 - 30xy \\ y - 1 & x + 1 \end{bmatrix}$$ Ans

Computing the Jacobian matrix using chain rule :

$$DF(x,y,z) = \begin{bmatrix} \frac{\partial}{\partial x}(3xy) & \frac{\partial}{\partial y}(3xy) & \frac{\partial}{\partial z}(3xy) \\ \frac{\partial}{\partial x}(y-z) & \frac{\partial}{\partial y}(y-z) & \frac{\partial}{\partial z}(y-z) \end{bmatrix}$$

$$DF(x,y,z) = \begin{bmatrix} 3y & 3x & 0 \\ 0 & 1 & -1 \end{bmatrix}$$

$$DG(x,y) = \begin{bmatrix} \frac{\partial}{\partial x}(x-5y) & \frac{\partial}{\partial y}(x-5y) \\ \frac{\partial}{\partial x}(xy) & \frac{\partial}{\partial y}(xy) \\ \frac{\partial}{\partial x}(x-y) & \frac{\partial}{\partial y}(x-y) \end{bmatrix}$$

$$DG(x,y) = \begin{bmatrix} 1 & -5 \\ y & x \\ 1 & -1 \end{bmatrix}$$

Using the chain rule :-

$$DF(G(x,y)) = \begin{bmatrix} 3(xy) & 3(x-5y) & 0 \\ 0 & 1 & -1 \end{bmatrix}$$

↑ Compute DF then plug in $G(x,y)$

$$= \begin{bmatrix} 3xy & 3x-15y & 0 \\ 0 & 1 & -1 \end{bmatrix}$$

Using the chain rule :-

$$D(F \circ G)(x,y) = DF(G(x,y)) * DG(x,y)$$

$$= \underset{2\times 3}{\begin{bmatrix} 3xy & 3x-15y & 0 \\ 0 & 1 & -1 \end{bmatrix}} \underset{3\times 2}{\begin{bmatrix} 1 & -5 \\ y & x \\ 1 & -1 \end{bmatrix}}$$

$$= \begin{bmatrix} 3xy + 3xy - 15y^2 & -15xy + 3x^2 - 15xy \\ y-1 & x+1 \end{bmatrix}$$

$$= \underset{2\times 2}{\begin{bmatrix} 6xy - 15y^2 & 3x^2 - 30xy \\ y-1 & x+1 \end{bmatrix}}$$ Ans

2.11>. Explain the need to order nodes when traversing the computation graph in forward and backward passes.

Soln :- Here, I need to compute the nodes preceding me before I move to the next node.

Due to this reason, it is needed to order nodes when traversing the computation graph in forward and backward passes.

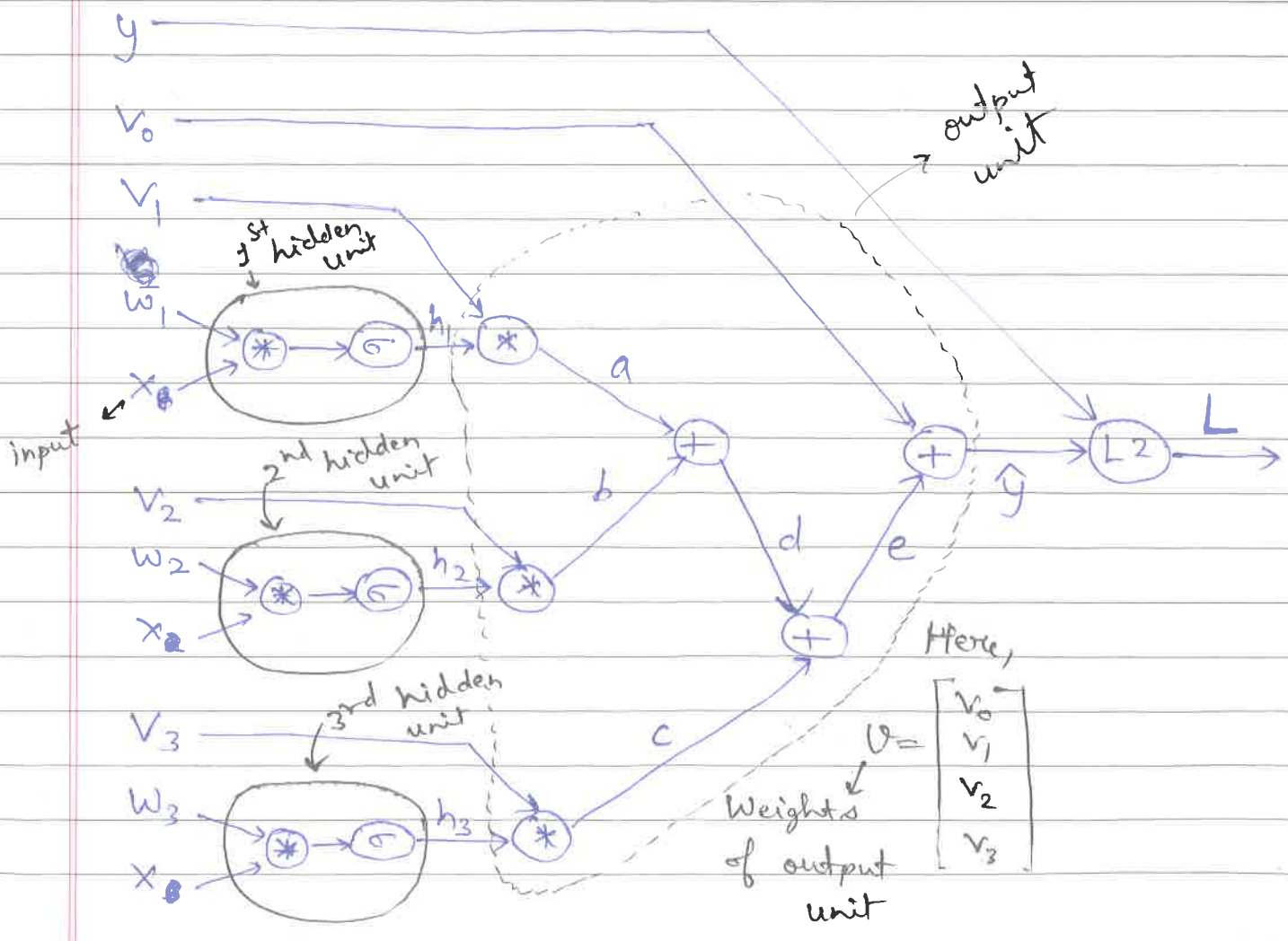
Q.12) Scalar computation graph:

(a). Draw the computation graph for a regression network with three hidden units and one output unit. Assume logistic activation at hidden layer nodes and linear activation at output. Assume L2 loss.

Soln:-

$$L = L_2\Big(\overbrace{V_0 + \sigma(XW_1)V_1 + \sigma(XW_2)V_2 + \sigma(XW_3)V_3}^{\hat{y}}, y\Big)$$

$$L = \Big(\big(V_0 + \sigma(XW_1)V_1 + \sigma(XW_2)V_2 + \sigma(XW_3)V_3\big) - y\Big)^2$$



(b). Assume the current weight parameters guess is given by:

$$w_1 = (0.01, 0.02, 0.03)$$
$$w_2 = (0.03, 0.01, 0.02)$$
$$w_3 = (0.02, 0.03, 0.01)$$
$$v = (0.01, 0.02, 0.03, 0.04)$$

where, w_1, w_2, w_3 are hidden unit weight vectors and v is the output unit weight vector. Assume the training data in the batch is given by:

$$x_1, y_1 = [(1,2), 8]$$
$$x_2, y_2 = [(1,3), 11]$$
$$x_3, y_3 = [(2,2), 10]$$

Compute the values at the network for the forward pass for each data point.

Soln :- ⇒ For $x_1, y_1 = [(1,2), 8]$

$$h_1 = \sigma(w_1 x_1)$$
$$= \sigma(0.01 + 0.02 \times 1 + 0.03 \times 2)$$
$$= \sigma(0.09)$$
$$= 0.52248.$$

$h_2 = \sigma(w_2 x_1)$
$= \sigma(0.03 + 0.01 \times 1 + 0.02 \times 2)$
$= \sigma(0.08)$
$= 0.5199$.

$h_3 = \sigma(w_3 x_1)$
$= \sigma(0.02 + 0.03 \times 1 + 0.01 \times 2)$
$= \sigma(0.07)$
$= 0.51749$.

$a = v_1 \times h_1$
$= 0.02 \times 0.52248$
$= 0.01049$.

$b = v_2 \times h_2$
$= 0.03 \times 0.5199$
$= 0.01559$.

$c = v_3 \times h_3$
$= 0.04 \times 0.51749$
$= 0.02069$.

$d = a + b$
$= 0.01049 + 0.01559$
$= 0.02608$.

$e = d + c$
$= 0.02069 + 0.02608$
$= 0.04677$.

$\hat{y} = v_0 + e$
$= 0.01 + 0.04677$
$= 0.05677$.

$$L = (\hat{y} - y)^2$$
$$= (0.05677 - 8)^2$$
$$= 63.09$$

$\Rightarrow$ For $x_2, y_2 = [(1,3), 1]$

$$h_1 = \sigma(w_1 x_2)$$
$$= \sigma(0.01 + 0.02 \times 1 + 0.03 \times 3)$$
$$= \sigma(0.12)$$
$$= 0.52996$$

$$h_2 = \sigma(w_2 x_2)$$
$$= \sigma(0.03 + 0.01 \times 1 + 0.02 \times 3)$$
$$= \sigma(0.1)$$
$$= 0.52497$$

$$h_3 = \sigma(w_3 x_2)$$
$$= \sigma(0.02 + 0.03 \times 1 + 0.01 \times 3)$$
$$= \sigma(0.08)$$
$$= 0.5199$$

$$a = v_1 \times h_1$$
$$= 0.02 \times 0.52996$$
$$= 0.0106$$

$$b = v_2 \times h_2$$
$$= 0.03 \times 0.52497$$
$$= 0.0157491$$

$$c = v_3 \times h_3$$
$$= 0.04 \times 0.5199$$
$$= 0.020796 .$$

$$d = a + b$$
$$= 0.0106 + 0.0157491$$
$$= 0.0263 .$$

$$e = d + c$$
$$= 0.0263 + 0.0208$$
$$= 0.0471 .$$

$$\hat{y} = v_0 + e$$
$$= 0.01 + 0.0471$$
$$= 0.0571 .$$

$$L = (\hat{y} - y)^2$$
$$= (0.0571 - 11)^2$$
$$= 119.7 .$$

$\Rightarrow$ For $x_3, y_3 = [(2,2), 10]$

$$h_1 = \sigma(W_1 x_3)$$
$$= \sigma(0.01 + 0.02 \times 2 + 0.03 \times 2)$$
$$= \sigma(0.11)$$
$$= 0.5275 .$$

$$h_2 = \sigma(W_2 x_3)$$
$$= \sigma(0.03 + 0.01 \times 2 + 0.02 \times 2)$$
$$= \sigma(0.09)$$
$$= 0.5225 .$$

$h_3 = \sigma(W_3 X_3)$

$= \sigma(0.02 + 0.03 \times 2 + 0.01 \times 2)$

$= \sigma(0.1)$

$= 0.525$.

$a = v_1 \times h_1$

$= 0.02 \times 0.5275$

$= 0.01055$.

$b = v_2 \times h_2$

$= 0.03 \times 0.5225$

$= 0.01568$

$c = v_3 \times h_3$

$= 0.04 \times 0.525$

$= 0.021$.

$d = a + b$

$= 0.01055 + 0.01568$

$= 0.02623$.

$e = d + c$

$= 0.02623 + 0.021$

$= 0.04723$.

$\hat{y} = v_0 + e$

$= 0.01 + 0.04723$

$= 0.05723$.

$L = (\hat{y} - y)^2$

$= (0.05723 - 10)^2$

$= 98.8$.

(c). Compute the gradients ~~using the formula~~ of the loss function with respect to w_1, w_2, w_3, v due to all training examples using back-propagation.

Soln :- ~~For $x_1, y_1 = [(1,2), 8]$~~

$$\frac{\partial L}{\partial \hat{y}} = 2(\hat{y} - y) \qquad \frac{\partial e}{\partial d} = 1 \qquad \frac{\partial d}{\partial b} = 1$$

$$\frac{\partial \hat{y}}{\partial e} = 1 \qquad \frac{\partial d}{\partial a} = 1 \qquad \frac{\partial e}{\partial c} = 1$$

$$\frac{\partial a}{\partial h_1} = v_1 \qquad \frac{\partial b}{\partial h_2} = v_2 \qquad \frac{\partial c}{h_3} = v_3$$

$$\frac{\partial h_1}{\partial w_1} = \sigma(w_1 x)(1 - \sigma(w_1 x)) \times$$

$$\frac{\partial h_2}{\partial w_2} = \sigma(w_2 x)(1 - \sigma(w_2 x)) \times$$

$$\frac{\partial h_3}{\partial w_3} = \sigma(w_3 x)(1 - \sigma(w_3 x)) \times$$

$$\frac{\partial \hat{y}}{\partial v_0} = 1 \qquad \frac{\partial a}{\partial v_1} = h_1 = \sigma(w_1 x)$$

$$\frac{\partial b}{\partial v_2} = h_2 = \sigma(w_2 x)$$

$$\frac{\partial c}{\partial v_3} = h_3 = \sigma(w_3 x)$$

$$\frac{\partial L}{\partial v_0} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial v_0} = 2(\hat{y}-y) * 1$$

$$\frac{\partial L}{\partial v_1} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial e} \frac{\partial e}{\partial d} \frac{\partial d}{\partial a} \frac{\partial a}{\partial v_1}$$

$$= 2(\hat{y}-y) * 1 * 1 * 1 * \sigma(w_1 x)$$

$$= 2(\hat{y}-y)\,\sigma(w_1 x)$$

$$\frac{\partial L}{\partial v_2} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial e} \frac{\partial e}{\partial d} \frac{\partial d}{\partial b} \frac{\partial b}{\partial v_2}$$

$$= 2(\hat{y}-y) \cdot 1 \cdot 1 \cdot 1 \cdot \sigma(w_2 x)$$

$$= 2(\hat{y}-y)\,\sigma(w_2 x)$$

$$\frac{\partial L}{\partial v_3} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial e} \frac{\partial e}{\partial c} \frac{\partial c}{\partial v_3}$$

$$= 2(\hat{y}-y) \cdot 1 \cdot 1 \cdot \sigma(w_3 x)$$

$$= 2(\hat{y}-y)\,\sigma(w_3 x)$$

$$\frac{\partial L}{\partial V} = \begin{bmatrix} \frac{\partial L}{\partial v_0} \\ \frac{\partial L}{\partial v_1} \\ \frac{\partial L}{\partial v_2} \\ \frac{\partial L}{\partial v_3} \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{m} 2(\hat{y}^{(i)} - y^{(i)}) \\ \sum_{i=1}^{m} 2(\hat{y}^{(i)} - y^{(i)})\,\sigma(w_1 x^{(i)}) \\ \sum_{i=1}^{m} 2(\hat{y}^{(i)} - y^{(i)})\,\sigma(w_2 x^{(i)}) \\ \sum_{i=1}^{m} 2(\hat{y}^{(i)} - y^{(i)})\,\sigma(w_3 x^{(i)}) \end{bmatrix}$$

Ans.

$$\frac{\partial L}{\partial w_1} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial e} \frac{\partial e}{\partial d} \frac{\partial d}{\partial a} \frac{\partial a}{\partial h_1} \frac{\partial h_1}{\partial w_1}$$

$$= 2(\hat{y}-y) \cdot 1 \cdot 1 \cdot 1 \cdot v_1 \cdot \sigma(w_1 x)(1-\sigma(w_1 x)) x$$

$$\boxed{\frac{\partial L}{\partial w_1} = \sum_{i=1}^{m} 2(\hat{y}^{(i)}-y^{(i)}) \cdot v_1 \cdot \sigma(w_1 x^{(i)})(1-\sigma(w_1 x^{(i)})) x^{(i)}} \quad \text{Ans}$$

$$\frac{\partial L}{\partial w_2} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial e} \frac{\partial e}{\partial d} \frac{\partial d}{\partial b} \frac{\partial b}{\partial h_2} \frac{\partial h_2}{\partial w_2}$$

$$= 2(\hat{y}-y) \cdot 1 \cdot 1 \cdot 1 \cdot v_2 \cdot \sigma(w_2 x)(1-\sigma(w_2 x)) x$$

$$\boxed{\frac{\partial L}{\partial w_2} = \sum_{i=1}^{m} 2(\hat{y}^{(i)}-y^{(i)}) \cdot v_2 \cdot \sigma(w_2 x^{(i)})(1-\sigma(w_2 x^{(i)})) x^{(i)}} \quad \text{Ans}$$

$$\frac{\partial L}{\partial w_3} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial e} \frac{\partial e}{\partial c} \frac{\partial c}{\partial h_3} \frac{\partial h_3}{\partial w_3}$$

$$= 2(\hat{y}-y) \cdot 1 \cdot 1 \cdot v_3 \cdot \sigma(w_3 x)(1-\sigma(w_3 x)) x$$

$$\boxed{\frac{\partial L}{\partial w_3} = \sum_{i=1}^{m} 2(\hat{y}^{(i)}-y^{(i)}) \cdot v_3 \cdot \sigma(w_3 x^{(i)})(1-\sigma(w_3 x^{(i)})) x^{(i)}} \quad \text{Ans}$$

$$\boxed{\frac{\partial L}{\partial V} = 2\sum_{i=1}^{m} (\hat{y}^{(i)}-y^{(i)}) z^{(i)}} \quad \text{Here, } z^{(i)} = \begin{bmatrix} 1 \\ \sigma(w_1 x^{(i)}) \\ \sigma(w_2 x^{(i)}) \\ \sigma(w_3 x^{(i)}) \end{bmatrix}$$

Q(d). Compute the gradients using the formula given in class and compare to your previous results.

Soln :-

$$\frac{\partial L}{\partial V} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial V}$$

$$\boxed{\frac{\partial L}{\partial V} = 2 \sum_{i=1}^{m} (\hat{y}^{(i)} - y^{(i)}) \, z^{(i)}}$$

$$\frac{\partial L}{\partial w_j} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial z_j} \frac{\partial z_j}{\partial w_j}$$

no. of units in hidden layer.

$$= 2 \sum_{i=1}^{m} (\hat{y}^{(i)} - y^{(i)}) \, v_j \cdot z_j^{(i)} (1 - z_j^{(i)}) \, x^{(i)}$$

So,

$$\boxed{\frac{\partial L}{\partial w_1} = 2 \sum_{i=1}^{m} (\hat{y}^{(i)} - y^{(i)}) . v_1 \cdot z_1^{(i)} (1 - z_1^{(i)}) \, x^{(i)}}$$

$$\boxed{\frac{\partial L}{\partial w_2} = 2 \sum_{i=1}^{m} (\hat{y}^{(i)} - y^{(i)}) . v_2 \cdot z_2^{(i)} (1 - z_2^{(i)}) \, x^{(i)}}$$

$$\boxed{\frac{\partial L}{\partial w_3} = 2 \sum_{i=1}^{m} (\hat{y}^{(i)} - y^{(i)}) . v_3 \cdot z_3^{(i)} (1 - z_3^{(i)}) \, x^{(i)}}$$

Here, $z_1^{(i)} = \sigma(w_1 x^{(i)})$; $z_2^{(i)} = \sigma(w_2 x^{(i)})$

$$Z_3^{(i)} = \sigma(W_3 x^{(i)})$$

On comparing with previous results, we came to know that both are same.

Q.13). Vector computation graph :-

(a). Let $f(x,y) = (2x+3y)^2$. Compute $\nabla f(x,y)$.

Soln :- $f(x,y) = (2x+3y)^2$

$$\nabla f(x,y) = \begin{bmatrix} \frac{\partial f(x,y)}{\partial x} \\ \frac{\partial f(x,y)}{\partial y} \end{bmatrix} = \begin{bmatrix} 2(2x+3y)\cdot 2 \\ 2(2x+3y)\cdot 3 \end{bmatrix}$$

$$\nabla f(x,y) = \begin{bmatrix} 8x+12y \\ 12x+18y \end{bmatrix}$$ Ans.

(b). Let $F(x,y) = \begin{bmatrix} x^2+2y \\ 3x+4y^2 \end{bmatrix}$. Compute the Jacobian matrix $DF(1,2)$.

Soln :- $F(x,y) = \begin{bmatrix} x^2+2y \\ 3x+4y^2 \end{bmatrix}$

$$DF(x,y) = \begin{bmatrix} \frac{\partial}{\partial x}(x^2+2y) & \frac{\partial}{\partial y}(x^2+2y) \\ \frac{\partial}{\partial x}(3x+4y^2) & \frac{\partial}{\partial y}(3x+4y^2) \end{bmatrix}$$

$$DF(x,y) = \begin{bmatrix} 2x & 2 \\ 3 & 8y \end{bmatrix}.$$

Now,

$$DF(1,2) = \begin{bmatrix} 2*1 & 2 \\ 3 & 8*2 \end{bmatrix}$$

$$\boxed{DF(1,2) = \begin{bmatrix} 2 & 2 \\ 3 & 16 \end{bmatrix}} \text{ Ans.}$$

(c). Let $G(x) = \begin{bmatrix} x \\ x^2 \end{bmatrix}$ and $F(x,y)$ as in the previous question. Compute the Jacobian matrix $D(F \circ G)(2)$ both with and without the chain rule.

Soln :- $G(x) = \begin{bmatrix} x \\ x^2 \end{bmatrix}$; $F(x,y) = \begin{bmatrix} x^2+2y \\ 3x+4y^2 \end{bmatrix}$

$$(F \circ G)(x) = \begin{bmatrix} x^2 + 2(x^2) \\ 3x + 4(x^2)^2 \end{bmatrix} = \begin{bmatrix} 3x^2 \\ 3x+4x^4 \end{bmatrix}.$$

Computing the Jacobian matrix directly :-

$$D(F \circ G)(x) = \begin{bmatrix} \frac{\partial}{\partial x}(3x^2) \\ \frac{\partial}{\partial x}(3x+4x^4) \end{bmatrix} = \begin{bmatrix} 6x \\ 3+16x^3 \end{bmatrix}$$

$$D(F \circ G)(2) = \begin{bmatrix} 6*2 \\ 3+16(2)^3 \end{bmatrix} = \begin{bmatrix} 12 \\ 3+128 \end{bmatrix}$$

$$\boxed{D(F \circ G)(2) = \begin{bmatrix} 12 \\ 131 \end{bmatrix}}$$ Ans

Computing the Jacobian matrix using chain rule :-

$$DF(x,y) = \begin{bmatrix} \frac{\partial}{\partial x}(x^2+2y) & \frac{\partial}{\partial y}(x^2+2y) \\ \frac{\partial}{\partial x}(3x+4y^2) & \frac{\partial}{\partial y}(3x+4y^2) \end{bmatrix}$$

$$DF(x,y) = \begin{bmatrix} 2x & 2 \\ 3 & 8y \end{bmatrix}$$

$$DG(x) = \begin{bmatrix} \frac{\partial}{\partial x}(x) \\ \frac{\partial}{\partial x}(x^2) \end{bmatrix} = \begin{bmatrix} 1 \\ 2x \end{bmatrix}$$

$$DF(G(x)) = \begin{bmatrix} 2(x) & 2 \\ 3 & 8(x^2) \end{bmatrix} = \begin{bmatrix} 2x & 2 \\ 3 & 8x^2 \end{bmatrix}$$

first compute DF and then plug in G(x)

Using the chain rule :-

$$D(F \circ G)(x) = DF(G(x)) * DG(x)$$

$$= \begin{bmatrix} 2x & 2 \\ 3 & 8x^2 \end{bmatrix}_{2\times 2} * \begin{bmatrix} 1 \\ 2x \end{bmatrix}_{2\times 1}$$

$$= \begin{bmatrix} 2x + 4x \\ 3 + 16x^3 \end{bmatrix}$$

$$= \begin{bmatrix} 6x \\ 3 + 16x^3 \end{bmatrix}$$

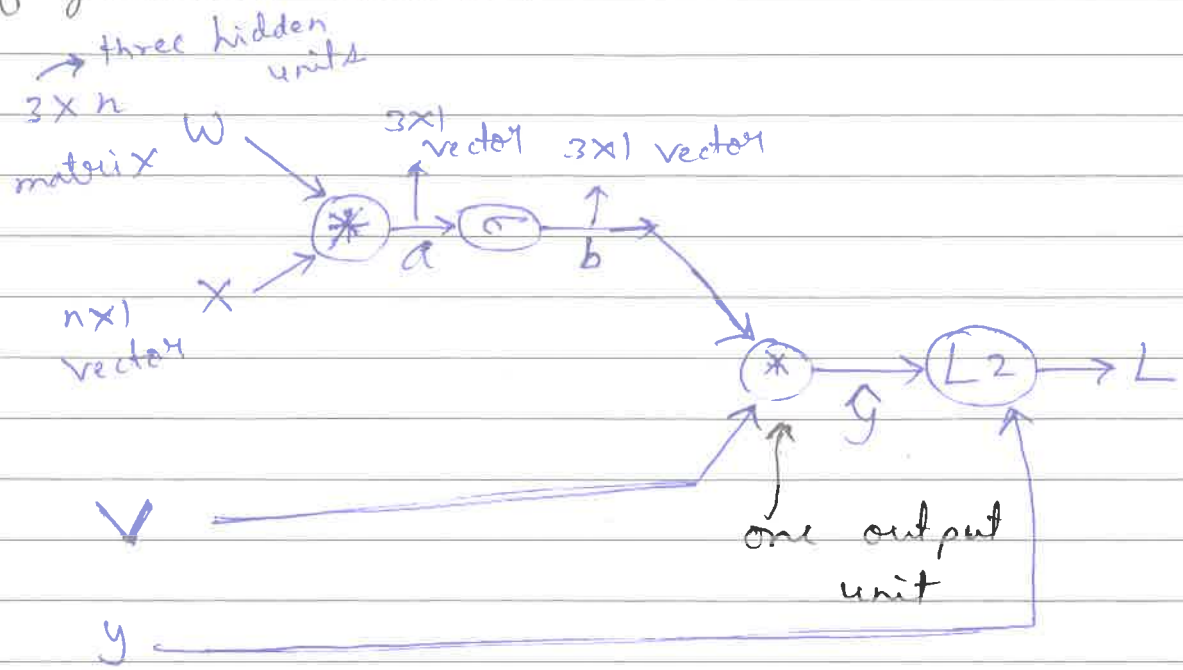
$$D(F \circ G)(2) = \begin{bmatrix} 6 * 2 \\ 3 + 16(2)^3 \end{bmatrix} = \begin{bmatrix} 12 \\ 3 + 128 \end{bmatrix}$$

$$\boxed{D(F \circ G)(2) = \begin{bmatrix} 12 \\ 131 \end{bmatrix}}$$ Ans.

(d). Repeat the scalar computation graph question (question 1) but this time using a vector notation. That is, draw the computation graph using vectors for input data and weights, perform a forward pass using vectors, and perform backpropagation by creating

and multiplying Jacobian matrices instead of gradient vectors.

Soln :-



forward pass :

$$a = Wx$$

$$b = \sigma(a) = \sigma(Wx)$$

$$\hat{y} = Vb$$

$$L = L_2(\hat{y}, y)$$

$$L = (\hat{y} - y)^2 .$$

backward pass :

$$\frac{\partial L}{\partial \hat{y}} = 2(\hat{y} - y) \qquad \frac{\partial b}{\partial a} = \sigma(a)(1 - \sigma(a))$$

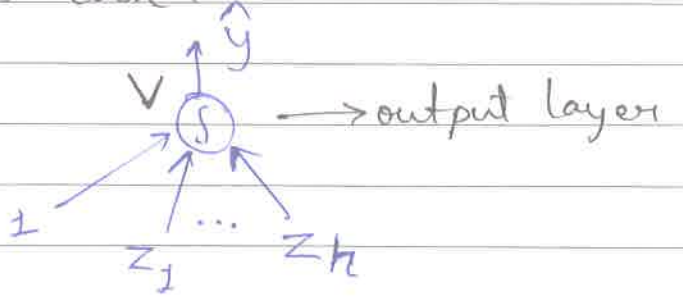
$$\frac{\partial \hat{y}}{\partial b} = V \qquad \frac{\partial a}{\partial W} = X .$$

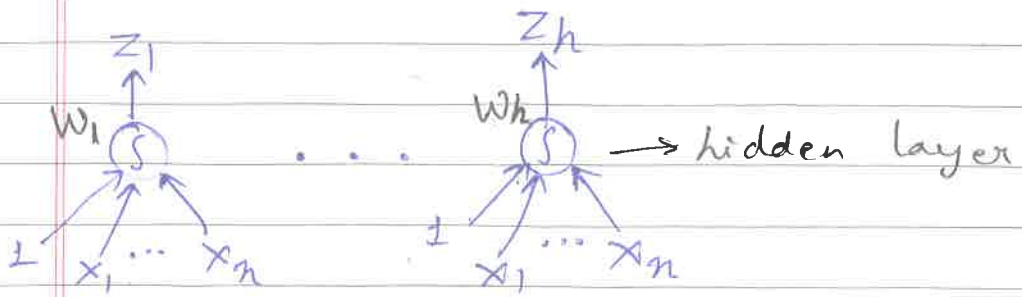
$$\frac{\partial L}{\partial w} = \frac{\partial L}{\partial \hat{y}} * \frac{\partial \hat{y}}{\partial b} * \frac{\partial b}{\partial a} * \frac{\partial a}{\partial w}$$

$$= 2(\hat{y} - y) * V * \sigma(a)(1 - \sigma(a)) * X$$

Q.14). Assume a two layer classification network with a single output where the hidden units and output unit use a Sigmoid activation. Assume a training set $\{x^{(i)}, y^{(i)}\}_{i=1}^{m}$ where $x^{(i)} \in R^n$ and $y^{(i)} \in \{0, 1\}$. Assume L2 loss $(\frac{1}{2}\sum_{i=1}^{m}(y^{(i)} - \hat{y}^{(i)})^2$ instead of binary cross entropy) derive the back propagation update equations for the output weights in a similar way to what was done in class for the binary cross entropy loss case.

Sol^{n}:-





Loss :-

$$\ell(\theta) = \frac{1}{2}\sum_{i=1}^{m}\left(\hat{y}^{(i)} - y^{(i)}\right)^2$$

Outputs :-

$$\hat{y} = \text{sigmoid}\left(V^T z\right)$$

$$z_j = \text{sigmoid}\left(w_j^T x\right)$$

Chain rule for output layer :-

$$\frac{\partial \ell(\theta)}{\partial V} = \frac{\partial \ell}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial V}$$

$$= \sum_{i=1}^{m} \frac{1}{2} \times 2 \left(\hat{y}^{(i)} - y^{(i)}\right) \hat{y}^{(i)} \left(1 - \hat{y}^{(i)}\right) z^{(i)}$$

$$= \sum_{i=1}^{m} \left(\hat{y}^{(i)} - y^{(i)}\right) \hat{y}^{(i)} \left(1 - \hat{y}^{(i)}\right) z^{(i)}$$

Chain rule for hidden layer :-

$$\frac{\partial \ell(\theta)}{\partial w_j} = \frac{\partial \ell}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial z_j} \frac{\partial z_j}{\partial w_j}$$

$$\frac{\partial \ell}{\partial w_j} = \sum_{i=1}^{m} \frac{1}{2} \times 2 \left(\hat{y}^{(i)} - y^{(i)}\right) V_j \, z_j^{(i)} \left(1 - z_j^{(i)}\right) x^{(i)}$$

$$\frac{\partial l}{\partial w_j} = \sum_{i=1}^{m} (\hat{y}^{(i)} - y^{(i)}) \cdot v_j \cdot z_j^{(i)} (1 - z_j^{(i)}) x^{(i)}$$

Update equation for output weights :-

$$V^{(new)} = V^{(old)} - \eta * \frac{\partial l}{\partial V}$$

(η : learning rate)

$$V^{(new)} = V^{(old)} - \eta * \sum_{i=1}^{m} (\hat{y}^{(i)} - y^{(i)}) \hat{y}^{(i)} (1 - \hat{y}^{(i)}) z^{(i)}$$

Ans

Update equation for hidden layer weights :-

$$w_{*j}^{(new)} = w_{*j}^{(old)} - \eta * \frac{\partial l}{\partial w_j}$$

$$w_j^{(new)} = w_j^{(old)} - \eta * \sum_{i=1}^{m} (\hat{y}^{(i)} - y^{(i)}) \cdot v_j \cdot z_j^{(i)} (1 - z_j^{(i)}) x^{(i)}$$

Ans