GRADIENT DECENT WITH BACKPROPAGATION

1. Initialize Network with Random Weight & Bias.

2. For each Train Image:

    a. Compute Activation for the Entire Network.

    b. Compute δ for Neurons in the Output Layer using Network Actual & Desired Activation:

    $$\delta_j^{(L)} = 2(\hat{a}_j^{(L)} - a_j^{(L)})\, a_j^{(L)}(1 - a_j^{(L)})$$

    c. Compute δ for all Neuron in the previous layer.

    $$\delta_j^{(l)} = \sum_K \delta_K^{(l+1)} w_{Kj}^{(l+1)} a_j^{(l)}(1 - a_j^{(l)})$$

    d. Compute Gradient of Cost w.r.t. each Weight & Bias for the training Image using δ:

    $$\frac{\partial C_x}{\partial w_{jK}^{(l)}} = \delta_j^{(l)} a_K^{(l-1)} \quad \Bigg| \quad \frac{\partial C_x}{\partial b_j^{(l)}} = \delta_j^{(l)}$$

3. Average the Gradient w.r.t. each Weight & Bias over the Entire Train Set:

$$\frac{\partial C}{\partial w_{jK}^{(l)}} = \frac{1}{n} \frac{\partial C_x}{\partial w_{jK}^{(l)}} \quad \Bigg| \quad \frac{\partial C}{\partial b_j^{(l)}} = \frac{1}{n} \sum \frac{\partial C_x}{\partial b_j^{(l)}}$$

4. Update the Weights & Biases using Gradient Descent

$$w_{jK}^{(l)'} = w_{jK}^{(l)} - \eta \frac{\partial C}{\partial w_{jK}^{(l)}} \quad \Bigg| \quad b_j^{(l)'} = b_j^{(l)} - \eta \frac{\partial C}{\partial b_j^{(l)}}$$

5. Repeat Step 2-4 till Cost is reduced below an acceptable level.

GRAD. Compute Complement.

(A) # of Multiplication req to Compute Cost of a Sing Image (C_x) : 23,820

(B) # of M. req to apply BP for a Sing Image : 24,210

(C) # of M. r. to Compute the Gradient of Cost for Single Image (∇C_x) : (A) + (B)
24,210 + 23820
= ~~2~~ 48,030

(D) # ~~of Train Images~~ of ~~for each~~ Train Images. : 60.000

# of Total # of Mult.. req. for One Iteration of Grad Descent : (C) x (D)
= 48,030 x 60,000
= 2.8 x 10^9

An improvement over Brute force by a factor of 10^4 10^4

Earlier it was 3.4×10^{13}

Image Identifier / ~~a~~ ~~bel~~ Labelling tagging

Gradient Computation Complexity

# of Multiplications req. to Compute Cost for Single Image (C_x)	: 23,820
# of Training Images used to Compute the Average Cost (C)	: 60,000
# of Multiplications req. to compute Average Cost (C)	: 23,820 x 60,000 1.4 x 10^9
# of Average Cost (C) Computations req. to Compute Gradient (∇C) w.r.t all Weights and Biases	: 23,861
Total # of Multiplications req. for One Iteration of Gradient Descent	: 23,861 x 1.4 x 10^9 3.4 x 10^{13}

Anmol :	500122363
Sumit .	5640
Ayush	4144
Abhishek .	500125001
Sheeba_Naaz	500117989
Abhay	500125918
Sachin :	500125994
Harshit :	500125256
Anurag :	500125693
Aditya :	500125257
Kartikey	500123293
Ankit :	500123760
Anuj :	500125988
Amul :	500124958
Harsh :	500125193
Hari	500125192
Rahul Katiyar	500126362
Ansh Verma	500126695
Manwar	500126361
Akshay	500127015
Anurag Vidhyarthi	500125910
Nikhil	500125751
Akshay verma	500120089
Asad Khan	500125189
Samy	500125581
Gurpreet Singh	500126791
Mangesh	500125990

Gradient Computation

using finite Difference

Parameter change Vector

$$\frac{\partial C}{\partial w_{jk}^{(\ell)}} \simeq \frac{C(w+\Delta w, b) - C(w,b)}{\Delta w_{jk}^{(\ell)}}, \quad \Delta w = \begin{bmatrix} 0 \\ \vdots \\ \Delta w_{jk}^{(\ell)} \\ \vdots \\ 0 \end{bmatrix}$$

$$\frac{\partial C}{\partial b_j^{(\ell)}} \simeq \frac{C(w, b+\Delta b) - C(w,b)}{\Delta b_j^{(\ell)}}; \quad \Delta b = \begin{bmatrix} 0 \\ \vdots \\ \Delta b_j^{(\ell)} \\ \vdots \\ 0 \end{bmatrix}$$

Init cost $C(w,b)$ is shared by all weights & bias

here

req one cost compute for each weight

Weight: $784 \times 30 + 30 \times 10 = 23820$

Bias: $30 + 10 = 40$

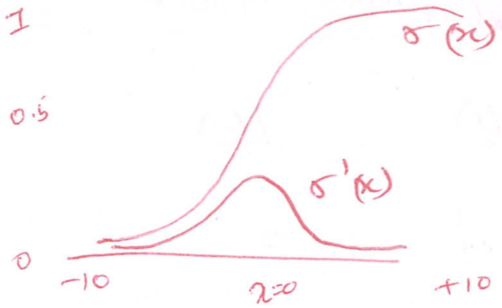
$$\left.\begin{array}{ll} \simeq 24{,}000 & \text{computation} \\ 60{,}000 & \text{train.} \end{array}\right\} 1.4 \times 10^9$$

Chhavi Kharb	500118635
Priyanka Mehra	500125114
Vansh	500120926
Soniya	500126096
Sakshi	500126523
Sidhant Gairola	500125992
Vedant Solanki	500121550
Tushar Pal	500118221
Renuka Dilore	500124565
Riya Lahiya	500117996
Rahul Singh	500123367
Rishabh Rawat	500125158
Devansh Pratap Singh	500125906
Akash Rawat	500123800
Khempal	500125754
SHIKHAR DWIVEDI	500125258
Sneha Verma	500125330

Gradient Descent with Backpropagation

1. Initialize Network with Random Weights and Biases
2. For each Training Image:
   a. Compute Activations for the Entire Network
   b. Compute δ for Neurons in the Output Layer using Network Activation and Desired Activation:
   $$\delta_j^{(L)} = 2\left(\hat{a}_j^{(L)} - a_j^{(L)}\right) a_j^{(L)} (1 - a_j^{(L)})$$
   c. Compute δ for all Neurons in the previous Layers:
   $$\delta_j^{(l)} = \sum_k \delta_k^{(l+1)} w_{kj}^{(l+1)} a_j^{(l)} (1 - a_j^{(l)})$$
   d. Compute Gradient of Cost w.r.t each Weight and Bias for the Training Image using δ:
   $$\frac{\partial C_x}{\partial w_{jk}^{(l)}} = \delta_j^{(l)} a_k^{(l-1)} \qquad \frac{\partial C_x}{\partial b_j^{(l)}} = \delta_j^{(l)}$$

Derava of Sigmoid Function



$$\sigma(z) = \frac{1}{1-e^{-z}}$$

its Derivati

$$\sigma'(z) = \frac{e^{-z}}{(1-e^{-z})^2}$$

$$\sigma'(z) = \sigma(z)\left[1-\sigma(z)\right]$$

For a Sigmoid neuron

$$a = a(z)$$

the $\sigma'(z) = a(1-a)$

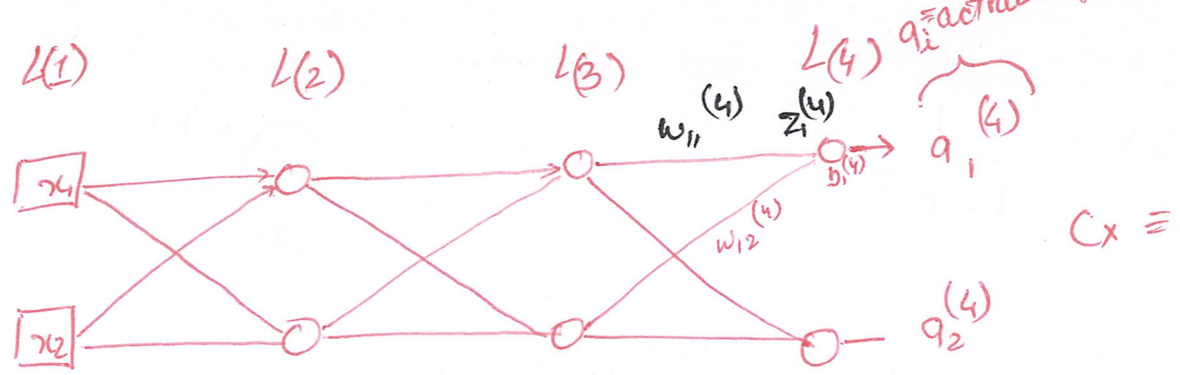
$$\frac{\partial C_2}{\partial w_{11}} = \underbrace{\left(2\left|\hat{a} - a_1^{(4)}\right|\right)\left(a_1^{(4)}\left(1-a_1^{(4)}\right)\right)}_{\text{Level } 4}\underbrace{\left(a_1^{3}\right)}_{\text{Level } 3}$$

$$\frac{\partial C_x}{\partial w_{11}} = \delta_1^{(4)}\, a_1^{(3)}$$

$$\frac{\partial C_x}{\partial w_{12}} = \delta_1^{(4)}\, a_2^{(3)}$$

$$\frac{\partial C_x}{\partial b_1^{(4)}} = \delta_1^{4} \cdot 1$$

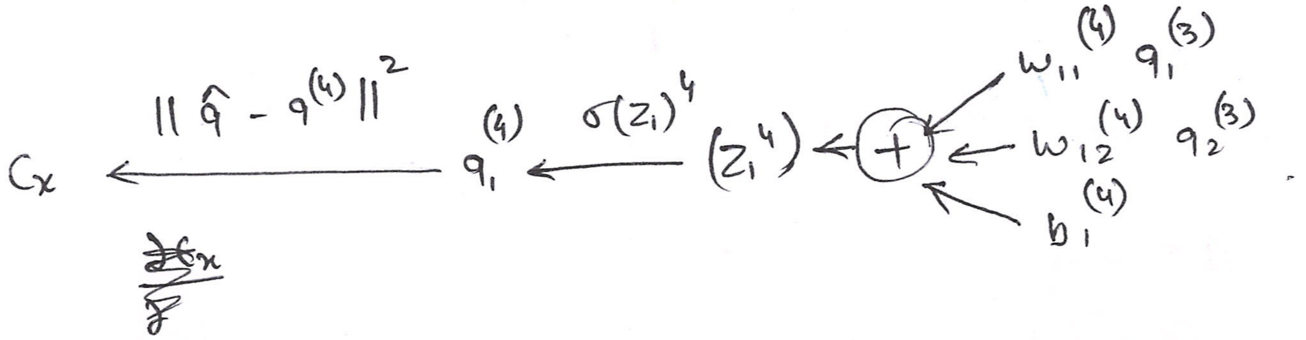
Computing Gradient using Chain Rule



$C_x \equiv$ cost

$\hat{a}$ is desired activation

$$C_x = \|\hat{a} - a^{(4)}\|^2$$



z_1^4 is Sum of weighted inputs

b_1^4 is bias of

$\sigma \equiv$ Sigmoid activation fn.

Let ~~you~~ of $w_{11}^{(4)}$

find derivative of cost wrt $w_{11}^{(4)} \equiv \dfrac{\partial C_x}{\partial w_{11}^{(4)}}$

By

$$\underbrace{\frac{\partial C_x}{\partial w_{11}^{(4)}} = \frac{\partial C_x}{\partial a_1^{(4)}} \cdot \frac{\partial a_1^{(4)}}{\partial z_1^{(4)}} \cdot \frac{\partial z_1^{(4)}}{\partial w_{11}^{(4)}}}_{\text{Chain Rule of derivatives}}$$

$$\frac{\partial C_x}{\partial a_1^{(4)}} = 2(\hat{a} - a_1^{(4)}) \quad \Bigg| \quad \frac{\partial a_1^{(4)}}{\partial z_1^{(4)}} = \sigma'(z_1^{(4)}) \quad \Bigg| \quad \frac{\partial z_1^{(4)}}{\partial w_{11}^{(4)}} = a_1^{(3)}$$

$$\therefore \frac{\partial C_x}{\partial w_{11}^{(4)}} = 2(\hat{a} - a_1^{(4)}) \cdot \sigma'(z_1^{(4)}) \cdot a_1^{(3)}$$

= all term are activations.

Back Propag of δ

For any neuro in the netw

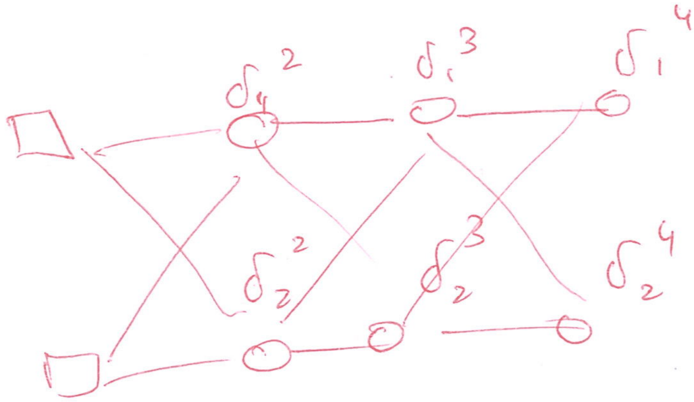
$$\delta_j^{(l)} = \sum_K \delta_K^{(l+1)} \quad a_j^{(l)} (1 - a_j^{(l)})$$

$\delta_j^{(l)}$: δ of j^{th} Neur in layer l

$\delta_j^{(l+1)}$: δ of K^{th} Neuro in layer $l+1$

$a_j^{(l)}$: Activat of j^{th} Neu in Lay l

$w_{Kj}^{(l+1)}$: Weig fro j^{th} Neuron in Lays l to K^{th} Neuro in layer $l+1$



$$\frac{\partial C_x}{\partial w_{21}^{(4)}} = \delta_2^{(4)} a_1^{(3)} \quad \Bigg| \quad \frac{\partial C_x}{\partial w_{22}^{(4)}} = \delta_2^{4} a_2^{(3)} \quad \Bigg| \quad \frac{\partial C_x}{\partial b_2^{4}} = \delta_2^{4} \cdot 1 \quad 3$$

1) 2nd ca