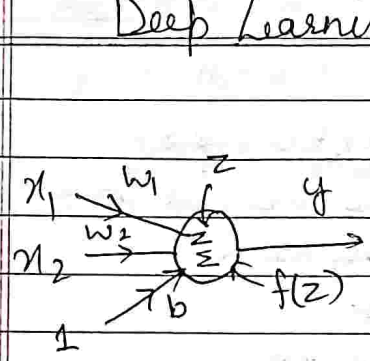
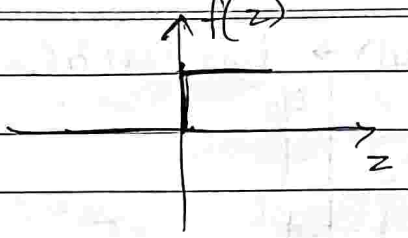
13/08 Deep Learning





$y = f(z)$

$$y = \begin{cases} 0, & f(z) \leq 0 \\ 1, & f(z) > 0 \end{cases}$$

$z = W^T x + b$

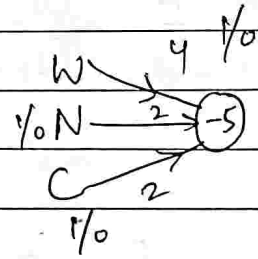
W = weights

b = bias

Linear classifier

More value of weight, more important is the feature

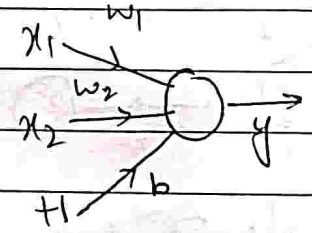
Eg: weather (W) — Binary
Near/far (N)
Comparsenion (C)



More importance is given to weather feature

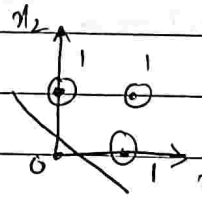
→ OR gate

x_1	x_2	o/p	
0	0	0	Class 1
0	1	1	Class 2
1	0	1	
1	1	1	



Line

$x_1 w_1 + x_2 w_2 + b = 0$



If $x_1 w_1 + x_2 w_2 + b \geq 0 \rightarrow y = 1$

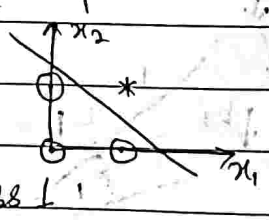
$\leq 0 \rightarrow y = 0$

for $w_1 = 1$, $w_2 = 1$, $b = 0$

x_1	x_2	z	y
0	0	0	0
0	1	1	1
1	1	2	1

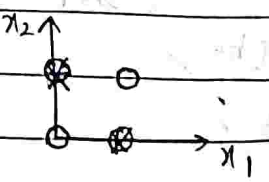
→ AND gate

x_1	x_2	o/p	
0	0	0	Class 1
0	1	0	
1	0	0	
1	1	1	Class 2



→ XOR

x_1	x_2	o/p	
0	0	0	C_1
0	1	1	C_2
1	0	1	C_2
1	1	0	C_1



DL

Page No.
Date

NAND → Universal gate

x_1	x_2	o/p
0	0	1
0	1	1
1	0	1
1	1	0

$x_1 w_1 + x_2 w_2 + b = z$

$z = 0w_1 + 0w_2 + b \Rightarrow z = b$

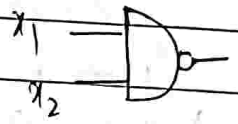
$z = w_1 + w_2 + b$

$z = w_1 + b$

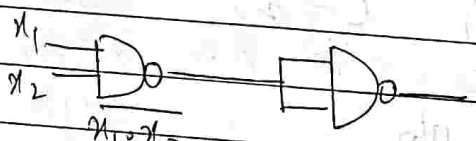
$z = w_1 + w_2 + b$

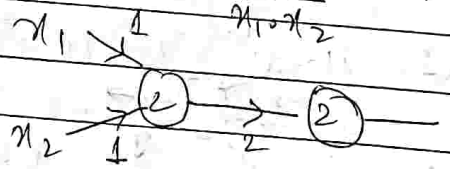
$w_1 = -1, \quad w_2 = -1, \quad b = 2$

x_1	x_2	o/p	z	y
0	0	1	2	1
0	1	1	1	1
1	0	1	1	1
1	1	0	0	0

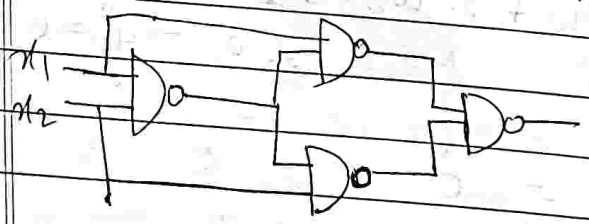


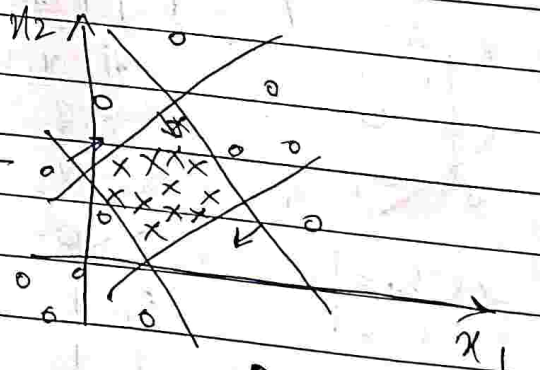
AND





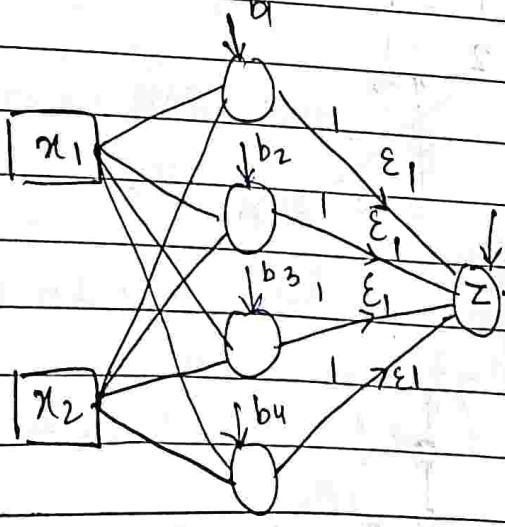
XOR from NAND

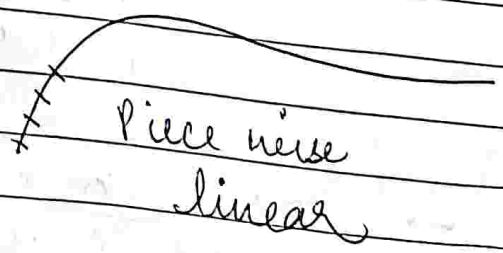




0 - class 1

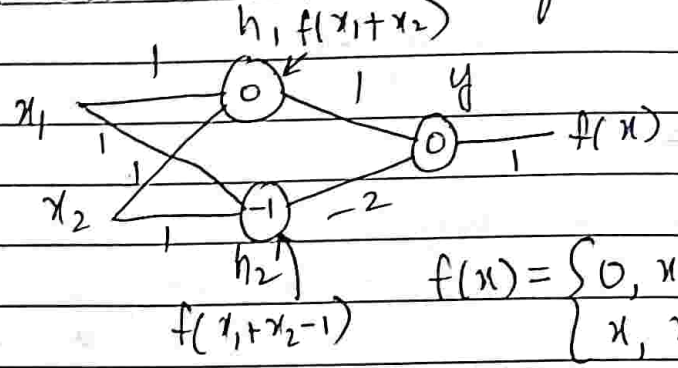
* - class 2

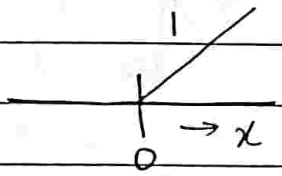




Page No.
Date

XOR with Relu activation function





0	0	0
0	1	1
1	0	1
1	1	0

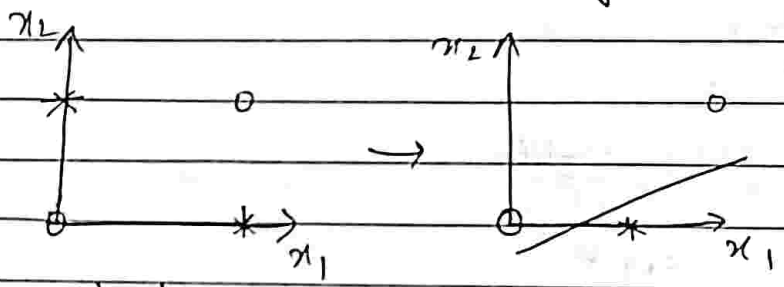
$$f(x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases}$$

$x_1\ x_2\ (0,0)$ $h_1\ h_2\ (0,0)$ $y = f(h_1 - 2h_2) = 0$

$x_1\ x_2\ (0,1)$ $h_1\ h_2\ (1,0)$ $y = f(1-0) = 1$

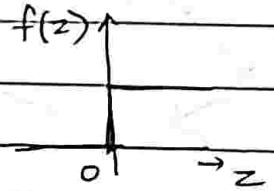
$x_1\ x_2\ (1,0)$ $h_1\ h_2\ (1,0)$ $y = f(1-0) = 1$

$x_1\ x_2\ (1,1)$ $h_1\ h_2\ (2,1)$ $y = f(2-2) = 0$



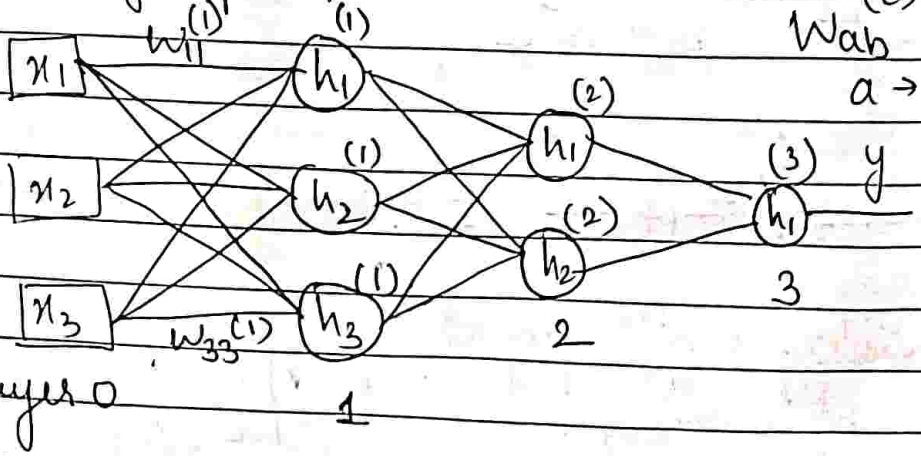
* Activation function

① Step function



② Sigmoid function

Multi layer perceptron



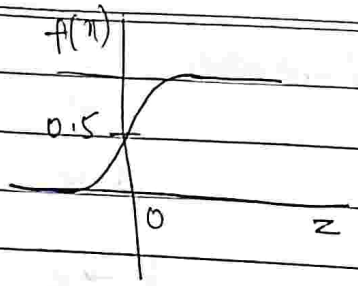
$W_{ab}^{(c)}$

a → coming from

b → going to

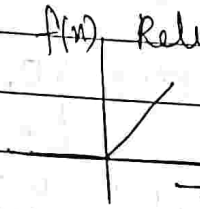
c → layer no.

Page No.
Date



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

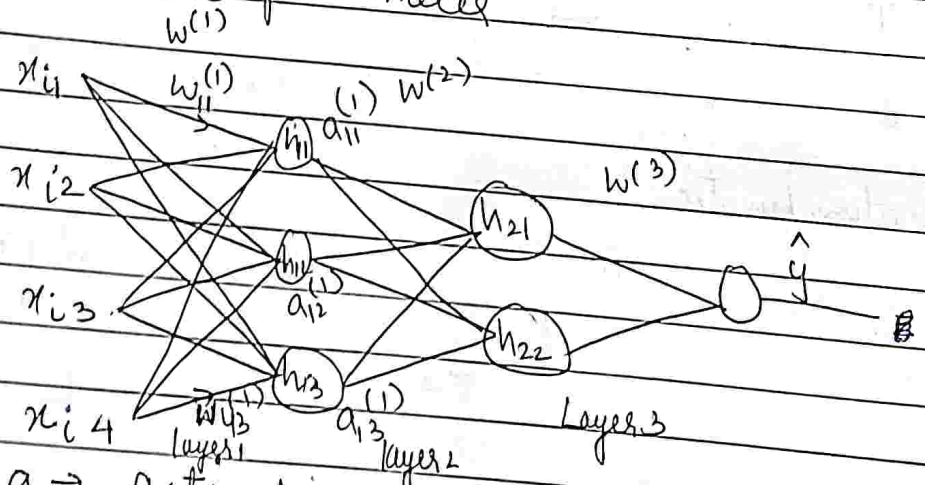
③ Relu function



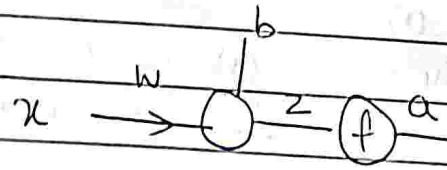
$$f(x) = \begin{cases} 0, & x \le 0 \\ x, & x > 0 \end{cases}$$

27/8 MLP

→ Feed forward propagation
→ loss
→ Back propagation
→ Activation
→ Trainable parameter



a → activation $z \to w^T x + b$ x → input
w → weight b → bias



$z = w^T x + b \qquad a = \sigma(z)$

$$z^{(1)} = W^{(1)T} x + b^{(1)}$$

$$z^{(1)} = \begin{bmatrix} w_{11}^{(1)} & w_{12}^{(1)} & w_{13}^{(1)} \\ w_{21}^{(1)} & w_{22}^{(1)} & w_{23}^{(1)} \\ w_{31}^{(1)} & w_{32}^{(1)} & w_{33}^{(1)} \\ w_{41}^{(1)} & w_{42}^{(1)} & w_{43}^{(1)} \end{bmatrix}^T \begin{bmatrix} x_{i1} \\ x_{i2} \\ x_{i3} \\ x_{i4} \end{bmatrix} + \begin{bmatrix} b_{11}^{(1)} \\ b_{12}^{(1)} \\ b_{13}^{(1)} \end{bmatrix}$$

$$a^{(1)} = \sigma(z^{(1)})$$

$(4\times3)^T \quad (4\times1) + (3\times1) =$

21/8 DL

$a^{(2)} = \sigma(z^{(2)})$ layer 2

$z^{(2)} = w^{(2)T} a^{(1)} + b^{(2)}$

$$z^{(2)} = \begin{bmatrix} w_{11}^{(2)} & w_{12}^{(2)} \\ w_{21}^{(2)} & w_{22}^{(2)} \\ w_{31}^{(2)} & w_{32}^{(2)} \end{bmatrix}^T \begin{bmatrix} a_{11}^{(1)} \\ a_{12}^{(1)} \\ a_{13}^{(1)} \end{bmatrix} + \begin{bmatrix} b_{11}^{(2)} \\ b_{12}^{(2)} \end{bmatrix} \qquad a^{(2)} = \sigma(z^{(2)})$$

$(3\times2)^T (3\times1) + (2\times1)$

$2\times1 + 2\times1 \rightarrow 2\times1 \rightarrow z^{(2)}$

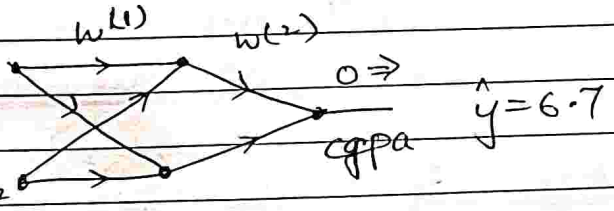
layer 3

$$z^{(3)} = \begin{bmatrix} w_{11}^{(3)} \\ w_{12}^{(3)} \end{bmatrix}^T \begin{bmatrix} a_{11}^{(2)} \\ a_{12}^{(2)} \end{bmatrix} + \begin{bmatrix} b_{11}^{(3)} \end{bmatrix}$$

$$\sigma\left(w^{(2)T}\left[\sigma\left(w^{(1)T} a^{(0)} + b^{(1)}\right)\right] + b\right)$$

- Loss functions

	x_1	x_2	class/value	
1	69	72	7	6.7
2			8	7.2
3			9	9.2



$L = y_i - \hat{y}_i$ = loss for single input

cost for entire dataset

loss = $f(w, b)$

→ Regression
- → Mean squared error (MSE)
- → Mean absolute error (MAE)
- → Huber loss

Classification
- → Binary cross entropy
- → Categorical cross entropy

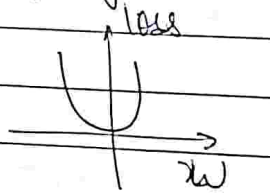
DL

Page No.
Date

- MSE (Mean Squared Error)

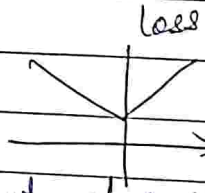
$$MSE = \frac{1}{n}\sum_{i=1}^{n}\underbrace{(y_i - \hat{y}_i)^2}_{e} \qquad \text{mean}(e^2)$$

$(y_i - \hat{y}_i)^2 \Rightarrow$ quadratic $\rightarrow$ differentiable $\rightarrow$ 1 local minima



More / drastic impact of outlier

- $MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$



Non differentiable

$\rightarrow$ low impact of outliers

29/8. • Huber loss

Parameter that decides whether it is an outlier or not

$$L = \begin{cases} \frac{1}{2}(y_i - \hat{y}_i)^2 ; \text{ if } |y_i - \hat{y}_i| \le \delta & \leftarrow \text{if } y_i \text{ is not outlier} \\ \delta|y_i - \hat{y}_i| - \delta^2 ; \text{ otherwise} & \leftarrow \text{when } y_i \text{ is outlier} \end{cases}$$

δ = trainable parameter

Decides whether datapoint y_i is ~~not~~ or not an outlier

MSE If $L = (y_i - \hat{y}_i)^2 \rightarrow$ cost $= \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$ Entire set

Single data point ↑ guide the n/w ↑ used for test & validation set

Training: data 1 → FP → loss → Update w, b → data 2 → FP → loss → Update w, b ← - - -

MAE $L = |y_i - \hat{y}_i|$

cost $= \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$

Page No.
Date

2918 DL

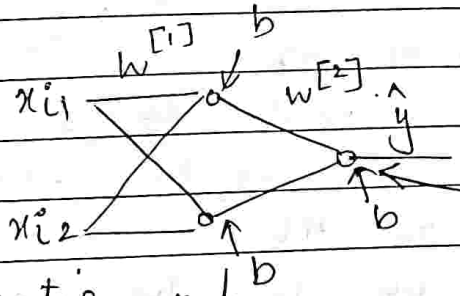
- Classification

1. Binary Cross Entropy

Eg:

x_1	x_2	y	$\hat{y}$
70	90	1	0.6
50	40	0	0.2
60	75	1	0.4

Eg:



For binary class classification only

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

$$\text{cost} = \frac{-1}{n}\sum_{i=1}^{n} y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)$$

For 1st set of input, $y_i = 1$ but $\hat{y}_i = 0.6$

$\Rightarrow L = -\log(0.6)$

2nd case $\rightarrow$ $y_i = 0$, $\hat{y}_i = 0.2$

$\Rightarrow L = -\log(0.8)$

3rd case $\rightarrow$ $y_i = 1$, $\hat{y}_i = 0.4 \Rightarrow L = -\log(0.6)$

2. Categorical Cross Entropy

For multiclass classification

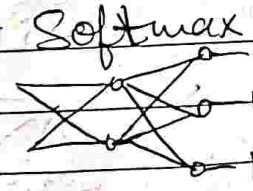
$$L = \sum_{i=1}^{K} -y_i \log(\hat{y}_i) \quad \text{where } K = \text{no. of classes}$$

Eg: If $K = 3 \Rightarrow L = -y_1 \log(\hat{y}_1) - y_2 \log \hat{y}_2 - y_3 \log \hat{y}_3$

Eg:

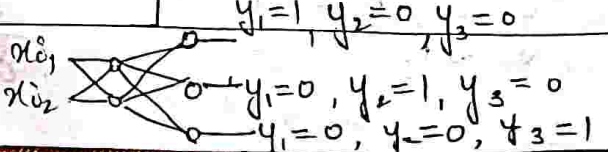
x_1	x_2	y	Ground truth OHE
70	90	Yes	[1 0 0]
60	40	No	[0 1 0]
40	40	Maybe	[0 0 1]
60	75	Yes	[1 0 0]

last activation has to be Softmax



$$\sum_{i=1}^{K} (f(z_i)) = 1$$

$$f(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$ activation for last layer

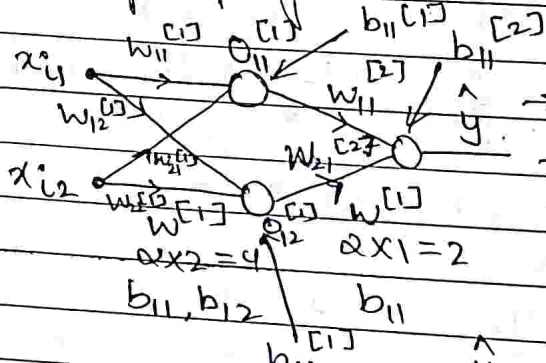


Page No.
Date

3. Sparse Categorical Entropy

x_1	x_2	y	OHE	Pred	L	O/P
70	90	Yes	1 0 0	[0.6, 0.2, 0.2]	$-\log 0.6$	1
60	40	No	0 1 0	[0.3, 0.5, 0.2]	$-\log 0.5$	2
40	40	Maybe	0 0 1			3
60	75	Yes	1 0 0			1

2/9/24 Backpropagation



→ Regression
→ Both activations linear
→ loss → MSE

$$w_{new} = w_{old} - \eta \frac{\partial L}{\partial w_{old}}$$

$$\hat{y} = f(O_{11}^{[1]} w_{11}^{[2]} + O_{12}^{[1]} w_{21}^{[2]} + b_{11}^{[2]})$$

For linear, $f(z) = z$

$$\Rightarrow \hat{y} = O_{11}^{[1]} w_{11}^{[2]} + O_{12}^{[1]} w_{21}^{[2]} + b_{11}^{[2]}$$

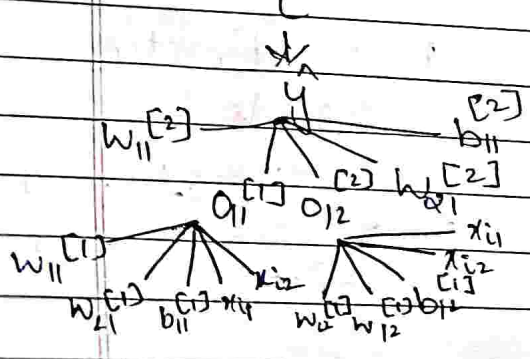
Initially, random weights and bias

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

$$\frac{\partial L}{\partial w_{11}^{[2]}} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial w_{11}^{[2]}}$$

$$\frac{\partial L}{\partial w_{21}^{[2]}} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial w_{21}^{[2]}}$$

$$\frac{\partial L}{\partial b_{11}^{[2]}} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial b_{11}^{[2]}}$$



$$\frac{\partial \hat{y}}{\partial w_{11}^{[2]}} = O_{11}^{[2]}$$

$$\Rightarrow \frac{\partial L}{\partial w_{11}^{[2]}} = -2(y - \hat{y}) O_{11}^{[2]}$$

$$\frac{\partial L}{\partial w_{21}^{[2]}} = -2(y - \hat{y}) O_{12}^{[1]}$$

To update $w_{21}^{[2]}$

$$\frac{\partial L}{\partial b_{11}^{[2]}} = -2(y - \hat{y}) \cdot 1$$

for $b_{11}^{[2]}$

2/9. DL

Page No.
Date

Backpropagation

$$w_{11}^{[2]}_{new} = w_{11}^{[2]}_{old} - \eta \frac{\partial L}{\partial w_{11}^{[2]}} \leftarrow \text{To update } w_{11}^{[2]}$$

$$O_{11}^{[1]} = f(x_{i1} w_{11}^{[1]} + x_{i2} w_{12}^{[1]} + b_{11}^{[1]})$$

$$O_{11}^{[1]} = x_{i1} w_{11}^{[1]} + x_{i2} w_{12}^{[1]} + b_{11}^{[1]} \quad \text{Here for linear}$$

$$\frac{\partial L}{\partial w_{11}^{[1]}} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial O_{11}^{[1]}} \cdot \frac{\partial O_{11}^{[1]}}{\partial w_{11}^{[1]}} =$$

$$\frac{\partial L}{\partial w_{21}^{[1]}} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial O_{11}^{[1]}} \cdot \frac{\partial O_{11}^{[1]}}{\partial w_{21}^{[1]}}$$

$$\frac{\partial L}{\partial w_{11}^{[1]}} = -2(y-\hat{y}) \cdot w_{11}^{[2]} \cdot x_{i1} \qquad \frac{\partial L}{\partial b_{11}^{[1]}} = -2(y-\hat{y}) w_{11}^{[2]}$$

$$\frac{\partial L}{\partial w_{21}^{[1]}} = -2(y-\hat{y}) \cdot w_{11}^{[2]} \cdot x_{i2}$$

$$O_{12}^{[1]} = x_{i1} w_{12}^{[1]} + x_{i2} w_{22}^{[1]} + b_{12}^{[1]}$$

$$\frac{\partial L}{\partial w_{12}^{[1]}} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial O_{12}^{[1]}} \frac{\partial O_{12}^{[1]}}{\partial w_{12}^{[1]}} = -2(y-\hat{y}) w_{21}^{[2]} x_{i1}$$

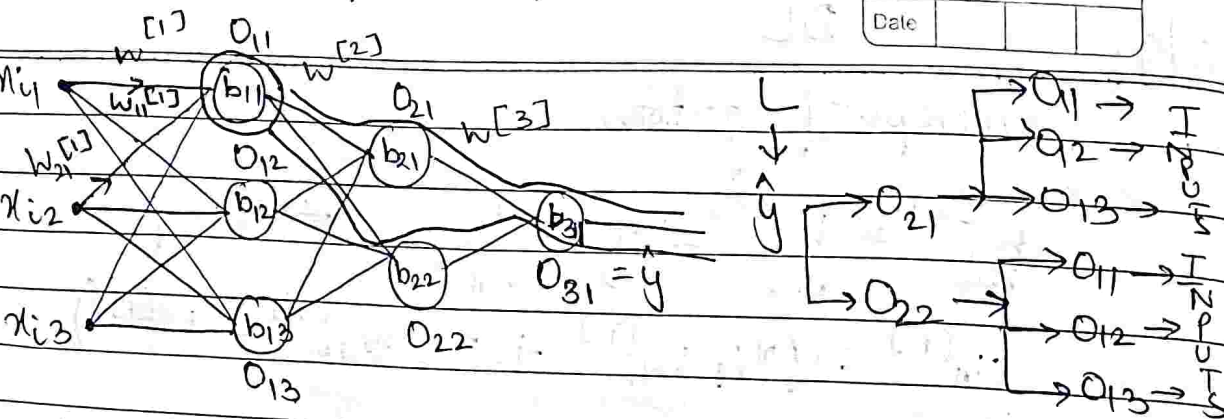
$$\frac{\partial L}{\partial w_{22}^{[1]}} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial O_{12}^{[1]}} \frac{\partial O_{12}^{[1]}}{\partial w_{22}^{[1]}} = -2(y-\hat{y}) w_{21}^{[2]} x_{i2}$$

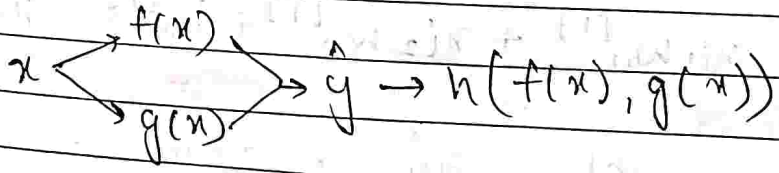
$$\frac{\partial L}{\partial b_{12}^{[1]}} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial O_{12}^{[1]}} \cdot \frac{\partial O_{12}^{[1]}}{\partial b_{12}^{[1]}} = -2(y-\hat{y}) w_{21}^{[2]}$$

2/9 → Lab → TSNE

3/9 Memorisation in MLP (Multilayer Perceptron)

DL



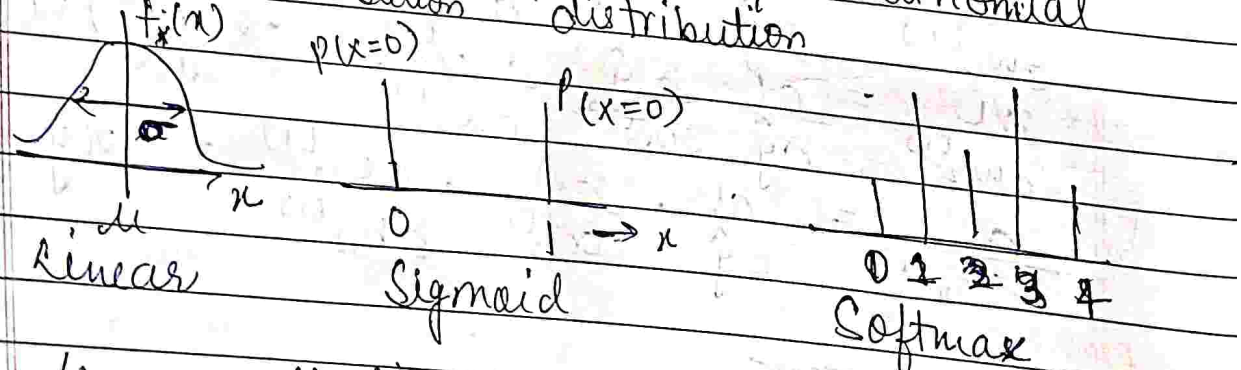


$$\frac{\partial L}{\partial w_{11}^{[1]}} = \frac{\partial L}{\partial \hat{y}} \left[\frac{\partial \hat{y}}{\partial O_{21}} \frac{\partial O_{21}}{\partial O_{11}} \frac{\partial O_{11}}{\partial w_{11}^{[1]}} + \frac{\partial \hat{y}}{\partial O_{22}} \cdot \frac{\partial O_{22}}{\partial O_{11}} \cdot \frac{\partial O_{11}}{\partial w_{11}^{[1]}} \right]$$

$$\frac{\partial L}{\partial w_{21}^{[1]}}$$

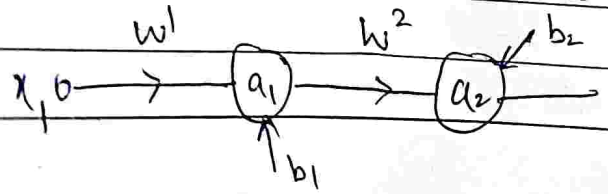
- Activation function
  - → hidden layer
  - → output layer
  - present at

Gaussian distribution | Bernoulli distribution | Multinoulli / multinomial distribution



1. Linear activation → $f(z) = z$

Linear combination of linear activations will result into linear plane.



$$z_1 = x_1 w^1 + b^1$$
$$a^1 = f(z_1) = x_1 w^1 + b^1$$

3/9 DL

$z_2 = a^1 w^{[2]} + b^{[2]} \rightarrow a^{[2]} = f(z_2) = z_2$

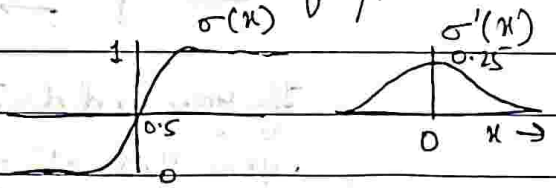
$a^{[2]} = [x_1 w^{[1]} + b^{[1]}] \cdot w^{[2]} + b^{[2]}$

$a^{[2]} = x_1 w^{[1]} w^{[2]} + b^{[1]} w^{[2]} + b^{[2]} = x_1 w + b$

9/9 Characteristics of activation function

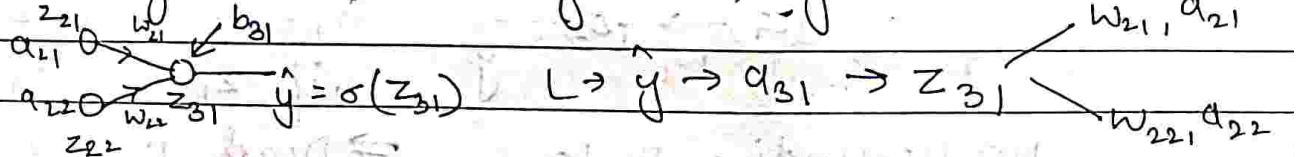
1. Non linear : Universal approximation theorem
2. Differentiable : weight update / back propogation
   exception = ReLU
3. Computationally inexpensive
4. Non saturating
5. Non centered zero : o/p of the activation should be centered to zero . mean of o/p is 0.

1. Sigmoid $\Rightarrow \sigma(x) = \frac{1}{1+e^{-x}} \quad \forall x$



Advantages → 1. Non linear 2. Differentiable 3. o/p is b/w [0, 1] consider o/p as ~~o/p~~ probabilities
4. Used for output layer

Disadvantages → 1. ~~Saturating~~ Saturating



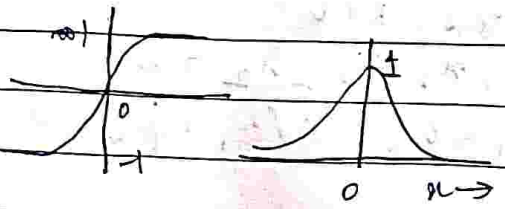
$z_{31} = a_{21} w_{21} + a_{22} w_{22} + b_{31}$

$\frac{\partial L}{\partial w_{21}} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial z_{31}} \cdot \frac{\partial z_{31}}{\partial w_{21}}$ (+ve)

$\frac{\partial L}{\partial w_{22}} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial z_{31}} \cdot \frac{\partial z_{31}}{\partial w_{22}}$ (+ve)

$w_{new} = w_{old} - \eta \frac{\partial L}{\partial w_{old}}$

2. Tanh ⇒

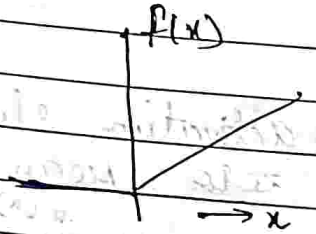


9/9 DL

Page No.
Date

DL lab
→ Grid search & Random search
No. of hidden layers
learning rate
activation fcn
Optimizer

10/9 • ReLU [Rectilinear Unit]

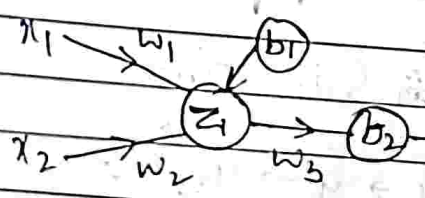


$f(x) = \max(0, x)$

Advantages
→ Non linear
→ computationally inexpensive

If you add 2 linear ReLU you get non linear

Disadvantages: O/p is not centered to zero as there are no negative value



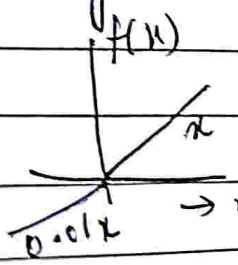
$z_1 = x_1 w_1 + x_2 w_2$
$a_1 = \max(0, z_1)$
if $z_1 < 0$, $a_1 = 0$

Not responding to changes ⇒ Dead neuron in input.

z_1 can be negative if η is high and if bias has high -ve value.

Dis → dead neurons and no response to -ve values

• Leaky ReLU

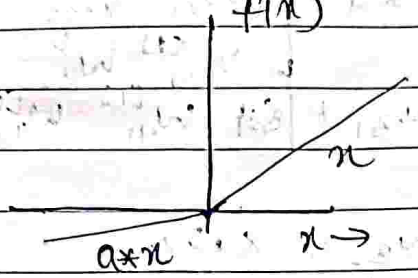


$$f'(x) = \begin{cases} 1, & x > 0 \\ 0.01, & x < 0 \end{cases}$$

$z_1 = w_1 x_1 + w_2 x_2 + b_1$
If $z_1 < 0$, $a(z_1) \neq 0$

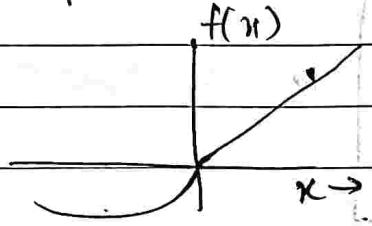
10/9 DL

- Parametric ReLU



a = hyperparameter or trainable parameter

- Exponential Linear Unit



$$f(x) = \begin{cases} x, & x > 0 \\ a(e^x - 1), & x < 0 \end{cases}$$

b/w 0.1 to 0.3

- Scalable Exponential Linear Unit

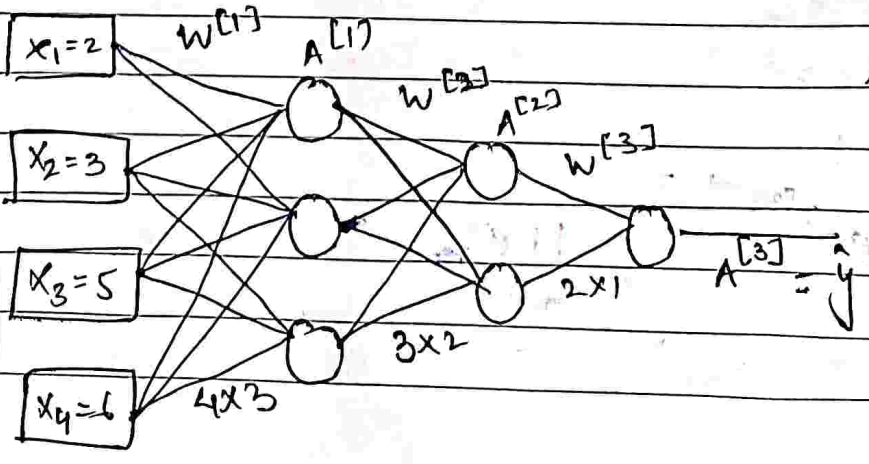
$$f(x) = \lambda \begin{cases} x, & x > 0 \\ a(e^x - 1), & x < 0 \end{cases} = \begin{cases} \lambda x, & x > 0 \\ \lambda a(e^x - 1), & x < 0 \end{cases}$$

12/9 Consider 2x2 image applied at the input of the following MLP. Perform forward propagation and weight update. Consider sigmoid activation on all neurons and all layers.

$$x = \begin{bmatrix} 2 & 3 \\ 5 & 6 \end{bmatrix} \quad W^{[1]} = \begin{bmatrix} 1 & 3 & -2 \\ 2 & -3 & 4 \\ 3 & -1 & 1 \\ 2 & 3 & -2 \end{bmatrix}, \quad W^{[2]} = \begin{bmatrix} 1 & -2 \\ -3 & 4 \\ -1 & 1 \end{bmatrix}$$

$$W^{[3]} = \begin{bmatrix} 4 \\ 1 \end{bmatrix}, \quad y = 1$$

Consider all bias terms are 0. and $\eta = 0.01$



$$x = [x_1 \; x_2 \; x_3 \; x_4]$$

Page No.
Date

$A^{[1]} = f^{[1]}(z^{[1]})$

$z^{[1]} = x w^{[1]} = [x_1\ x_2\ x_3\ x_4] \begin{bmatrix} w_{11}^{[1]} & w_{12}^{[1]} & w_{13}^{[1]} \\ w_{21}^{[1]} & w_{22}^{[1]} & w_{23}^{[1]} \\ w_{31}^{[1]} & w_{32}^{[1]} & w_{33}^{[1]} \\ w_{41}^{[1]} & w_{42}^{[1]} & w_{43}^{[1]} \end{bmatrix}$

$z_1^{[1]} = x_1 w_{11} + x_2 w_{21} + x_3 w_{31} + x_4 w_{41}$

$z_2^{[1]} = x_1 w_{12} + x_2 w_{22} + x_3 w_{32} + x_4 w_{42}$

$z_3^{[1]} = x_1 w_{13} + x_2 w_{23} + x_3 w_{33} + x_4 w_{43}$

$z^{[1]} = [2\ 3\ 5\ 6] \begin{bmatrix} 1 & 3 & -2 \\ 2 & -3 & 4 \\ 3 & -1 & 1 \\ 2 & 3 & -2 \end{bmatrix}$

$z_1^{[1]} = 2+6+15+12 = 35$

$z_2^{[1]} = 6+(-9)-5+18 = 10 \Rightarrow z^{[1]} = [35\ 10\ 1]$

$z_3^{[1]} = -4+12+5-12 = 1$

$A^{[1]} = \frac{1}{1+e^{-z^{[1]}}} = \left[\frac{1}{1+e^{-35}} \quad \frac{1}{1+e^{-10}} \quad \frac{1}{1+e^{-1}}\right]$

$A^{[1]} = \left[\frac{e^{35}}{e^{35}+1} \quad \frac{e^{10}}{e^{10}+1} \quad \frac{e}{1+e}\right] = [1 \quad 0.9 \quad 0.73]$

$A^{[2]} = f^{[2]}(z^{[2]}) \quad \& \quad z^{[2]} = A^{[1]} \cdot w^{[2]}$

$z^{[2]} = \underset{1\times3}{[1\ 0.9\ 0.73]} \begin{bmatrix} 1 & -2 \\ -3 & 4 \\ -1 & 1 \end{bmatrix}_{3\times2} \Rightarrow [1-2.7-0.73 \quad -2+3.6+0.73]$

$= [-2.43 \quad 2.33]$

$z_1^{[2]} = -2.43$

$z_2^{[2]} = 2.33$

$A^{[2]} = \frac{1}{1+e^{-z^{[2]}}} = \left[\frac{1}{1+e^{-(-2.43)}} \quad \frac{1}{1+e^{-2.33}}\right]$

$= \left[\frac{1}{1+e^{2.43}} \quad \frac{e^{2.33}}{1+e^{2.33}}\right] = \left[\frac{1}{1+11.35888} \quad \frac{10.27794}{11.27794}\right]$

$= [0.08 \quad 0.91]$

$A^{[3]} = f^{[3]}(z^{[3]})$ & $z^{[3]} = A^{[2]} w^{[3]}$

$z^{[3]} = \begin{bmatrix} 0.08 & 0.91 \end{bmatrix} \begin{bmatrix} 4 \\ 1 \end{bmatrix} = 0.32 + 0.91$
$= 1.23$

$A^{[3]} = \frac{1}{1+e^{-z^{[3]}}} = \frac{1}{1+e^{-1.23}} = 0.77$

$\hat{y} = 0.77$ and $y = 1$ $\Rightarrow \hat{y} - y = 0.77 - 1$
$= -0.23$

$L = -y \log(\hat{y}) - (1-y) \log(1-\hat{y})$

$w_n^{[3]} = w_0^{[3]} - \eta \frac{\partial L}{\partial w_0^{[3]}}$

$\frac{\partial L}{\partial w_0^{[3]}} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial z^{[3]}} \frac{\partial z^{[3]}}{\partial w_0^{[3]}}$

$\frac{\partial L}{\partial \hat{y}} = \frac{-y}{\hat{y}} - \frac{(1-y)}{(1-\hat{y})}(-1) = \frac{-y}{\hat{y}} + \frac{1-y}{1-\hat{y}}$

$\frac{\partial \hat{y}}{\partial z^{[3]}} = \frac{\partial \hat{y}}{\partial A^{[3]}} \Rightarrow \hat{y} = \sigma(z^{[3]}) = \frac{1}{1+e^{-z^{[3]}}}$

$\frac{\partial \hat{y}}{\partial z^{[3]}} = \sigma(z^{[3]})(1 - \sigma(z^{[3]}))$

$= \hat{y}(1-\hat{y})$

$\frac{\partial z^{[3]}}{\partial w_0^{[3]}} = A^{[2]}$

$\frac{\partial L}{\partial w_0^{[3]}} = -\left(\frac{y}{\hat{y}} - \frac{1-y}{1-\hat{y}}\right) \hat{y}(1-\hat{y}) A^{[2]}$

$= -[y(1-\hat{y}) - \hat{y}(1-y)] A^{[2]}$

$= [\hat{y}(1-y) - y(1-\hat{y})] A^{[2]}$

$= [\hat{y} - y\hat{y} - y + y\hat{y}] A^{[2]} = (\hat{y} - y) A^{[2]}$

$w_n^{[3]} = \begin{bmatrix} 4 \\ 1 \end{bmatrix} - 0.01(\hat{y} - y) A^{[2]}$

But Cap

DL by Amit Das Book for numericals

$$w_n^{[2]} = w_0^{[2]} - \eta \frac{\partial L}{\partial w_0^{[2]}}$$

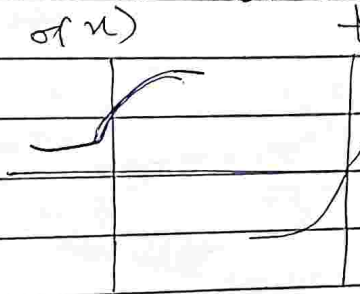
$$\frac{\partial L}{\partial w_0^{[2]}} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial A^{[2]}} \frac{\partial A^{[2]}}{\partial w_0^{[2]}}$$

17/9/24 Weight initialization

$W^{[1]}$ initial value of weight matrix

0 is not the correct way to initialize

He initialization & Xaviers initialization



i. $W^{[l]}$ = Normal random matrix

$x^T w$ $W^T x + b$

He 1) $W^{[l]}$ = Normal random matrix $* \sqrt{\frac{2}{size^{[l-1]}}}$

Xavier 2) $W^{[l]}$ = Uniform random matrix $* \sqrt{\frac{1}{size^{[l-1]}}}$

→ Gradient descent (GD)

$$w_n = w_0 - \eta \frac{\partial L}{\partial w_0}$$

→ Stochastic GD: epochs = n

for no. of epochs:

Shuffle

for data point

predict using forward prop.

loss

$\partial L / \partial w$

update weights

end

no of weight updates = epochs × data points

Take lot of times

Not guaranteed to reach somewhere or converge to near solution

→ Batch GD : no. of times weight updates = epochs
Faster

→ Mini batch : Data is randomly divided into patches
No. of batches × No. of epochs

→ Adaptive learning rate

$$w = w - \eta \frac{\partial L}{\partial w} \text{ (without momentum)}$$

$$mW_i = \beta_1^2 m\, W_{i-1} + (1-\beta_1)\frac{\partial L}{\partial w} \Rightarrow W_i = W_{i-1} - \eta m W_i$$

(With momentum)

• Adaptive Gradient (Adagrad)

$$\alpha_i = \frac{\alpha_{i-1}}{\sqrt{\delta + \sum_{k=1}^{i-1} \partial w_k^2}} \quad \delta \cong 10^{-5} \text{ to } 10^{-1}$$

25/9 DL

• Optimizer

$$\alpha_i = \frac{\alpha}{\sqrt{\delta + \sum_{k=1}^{i-1} \partial w_k^2}} \qquad w_i = w_{i-1} - \alpha_i \partial w$$

fixed 10^{-5} to 10^{-7}

• Root mean square propagation

More weightage is given to current gradient.

$$Sw_i = \beta_2 Sw_{i-1} + (1-\beta_2)\partial w_{i-1}^2$$

$$\alpha_i = \frac{\alpha \leftarrow \text{initial LR}}{\sqrt{\delta + Sw_i}}$$

• Gradient descent with momentum

$$mw_i = \beta_1 mw_{i-1} + (1-\beta_1)\frac{\partial L}{\partial w} \Rightarrow mw_i = \beta_1 mw_{i-1} + (1-\beta_1)\partial w$$

$$w_i = w_{i-1} - \alpha_i \partial w$$

$$w_i = w_{i-1} - \alpha mw_i = w_{i-1} \alpha [\beta_1 mw_{i-1} + (1-\beta_1)\partial w]$$

• Adam optimizer (AO)

$$\left.\begin{aligned} mw_i &= \beta_1 mw_{i-1} + (1-\beta_1)\partial w \\ w_i &= w_{i-1} - \alpha mw_i \end{aligned}\right] \text{GD with M}$$

AO

$$\begin{cases} Sw_i = \beta_2 Sw_{i-1} + (1-\beta_2)\partial w_i^2 \\ w_i = w_{i-1} - \alpha_i \partial w \\ mw_i^{\text{corrected}} = \dfrac{mw_i}{(1-\beta_1^i)} \end{cases} \quad \& \quad \alpha_i = \frac{\alpha}{\sqrt{\delta + Sw_i}}$$

$$Sw_i^{\text{corrected}} = \frac{Sw_{i-1}}{(1-\beta_2)^i}$$

$$\alpha_i^{\text{corrected}} = \frac{\alpha}{\sqrt{\delta + Sw_i^{\text{corrected}}}}$$

$$w_i = w_{i-1} - \alpha_i \, mw_i^{\text{corrected}}$$

Q For a NN, initial LR is 0.02 and squared derivate of weight in 1st iteration is 6. If SD of w ↑ by 5% in each iteration, find the LR of adagrad after 10 iterations. Assume $\delta = 10^{-5}$

$$\alpha_i = \frac{\alpha}{\sqrt{\delta + \partial w_i^2}}$$

$$\alpha_{10} = \frac{0.02}{\sqrt{10^{-5} + 6}} = \frac{0.02}{\sqrt{\frac{1}{10^5} + 6}} = \frac{0.02}{\sqrt{1 + 6\times 10^5}} = \frac{0.02}{174.5973} = \cancel{2.5819}$$

$$= 0.008$$

$\alpha_i =$

iteration	∂w^2	$\Sigma \partial w^2$	α_i
1	6	6	0.008
2	6.3	6+6.3=12.3	0.006
3	12.3×0.5+6 = 6.62	18.92	

1 → 6 → 6 → 0.08

2 → 6 + 6×0.05 = 6.3 → 6+6.3 = 12.3 →

3 → 12.3 + 12.3×0.05 (a) → 12.3 + a = → 25.219

12.915 → 12.915

$12.3 \times 0.05 + 6 \rightarrow 6.62$

iteration	∂w^2	$\Sigma \partial w^2$	α_i
1	6	6	0.008
2	6+6×0.05 = 6.3	6+6.3=12.3	0.006
3	12.3×0.05+6=6.62	12.3+6.62=18.92	
4	18.92×0.05+6=6.94	18.92+6.94=25.86	
5	25.86×0.05+6=7.293	7.293+25.86=33.153	
6	33.153×0.05+6=7.65	7.65+33.153=40.8	
7	40.8×0.05+6=8.04	8.04+40.8=48.84	
8	48.84×0.05+6=8.442	8.442+48.84 = 57.282	
9	57.282×0.05+6=8.8641	8.8641+57.282 = 66.14	
10	66.14×0.05+6 = 9.307	9.307+66.14 = 75.447	

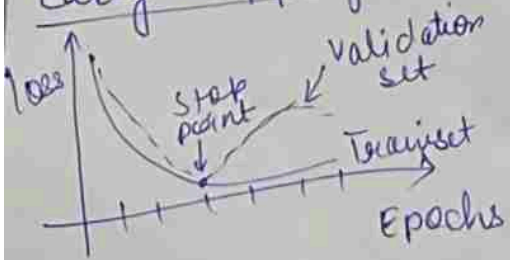
- **Regularization**

Types → L_1 & L_2

$L_1 \rightarrow L'(w,b) = L(w,b) + \lambda w_i$ ← sum of weights

$L_2 \rightarrow L'(w,b) = L(w,b) + \lambda w_i^2$ ← squared sum of weights

Early stopping



Dropout

Data augmentation

→ related to images
→ resize
→ rotate
→ shift

Normalization

$x_n = x_i - \mu x_i$

→ Normalize data by making it zero mean & 1 variance.

28/9 Unit 2 - CNN

0 - black, 255 black

Convolution : For input image $n \times n$ & filter $m \times n$, dimension of the output matrix is $(n-m+1) \times (n-m+1)$

Eg: 6x6 Image, 3x3 Filter $\Rightarrow$ Output $\rightarrow (6-3+1) \times (6-3+1)$ $\underline{4 \times 4}$

Padding & Stride 1/10

size of padding depends on size of filter

$n+2p-f+1 \rightarrow$ for padding

$= n \Rightarrow p = \frac{f-1}{2}$

With Stride $\rightarrow \frac{n-f}{s}+1$

without padding, $s \Rightarrow \frac{n-f}{s}+1 \times \frac{n-f}{s}+1$

with padding, $s \Rightarrow \frac{n+2p-f}{s}+1 \times \frac{n+2p-f}{s}+1$

padding $\rightarrow$ Valid $\rightarrow$ no padding

$\rightarrow$ Same $\rightarrow$ with padding o/p shape $\Rightarrow$ i/p shape

Q I/p image $\Rightarrow 28 \times 28$

layer 1, 2, 3 $\rightarrow$ 32 filter 3x3, no Stride, valid padding

layer 4 $\rightarrow$ Flatten

layer 5 $\rightarrow$ 128 neuron fully connected

6 $\rightarrow$ 10 neurons fully connected

Find o/p shape

$\rightarrow$ layer 1 = 26 x 26 x 32 (28 - 3 + 1 = 26)

layer 2 = 24 x 24 x 32 (26 - 3 + 1 = 24)

layer 3 = 22 x 22 x 32 (24 - 3 + 1 = 22)

layer 4 = 22 x 22 x 32

layer 5 = 128

layer 6 = 10

Q layer 1, 2, 3 $\rightarrow$ 32 filter 3x2, Stride = 2, same padding

1 $\rightarrow$ 14x14x32

2 $\rightarrow$ 7x7x32

3 $\rightarrow$ 4x4x32

4 $\rightarrow$ 4x4x32

5 $\rightarrow$ 128

6 $\rightarrow$ 10

layer 1 = 14x14x32 $\left(\frac{28+2-3}{2}+1\right)$

layer 2 = $\left(\frac{14+2-3}{2}+1\right) = 13.5+1 = 14$

$= \left(\frac{13}{2}+1\right) = 6.5+1 = 7$

$\rightarrow = 7 \times 7 \times 32$

layer 3 = $\left(\frac{7+2-3}{2}+1\right) = \left(\frac{6}{2}+1\right) = 3+1$

$= 4 \times 4 \times 32$

- Pooling

1. Image storage
2. Translation invariance

layer 2 = 24 x 24 x 32 (26 - 3 + 1 = 24)

layer 3 = 22 x 22 x 32 (24 - 3 + 1 = 22)

layer 4 = 22 x 22 x 32

layer 5 = 128

layer 6 = 10

a) layer 1, 2, 3 → 32 filter 3x2, Stride = 2, same padding

1 → 14x14x32

2 → 7x7x32

3 → 4x4x32

4 → 4x4x32

5 → 128

6 → 10

layer 1 = 14x14x32 $\left(\frac{28+2-3}{2}+1\right)$

layer 2 = $\left(\frac{14+2-3}{2}+1\right)$ = 13.5+1 = 14

$= \left(\frac{13}{2}+1\right) = 6.5+1 = 7$

→ = 7x7x32

layer 3 = $\left(\frac{7+2-3}{2}+1\right) = \left(\frac{6}{2}+1\right) = 3+1$

= 4x4x32

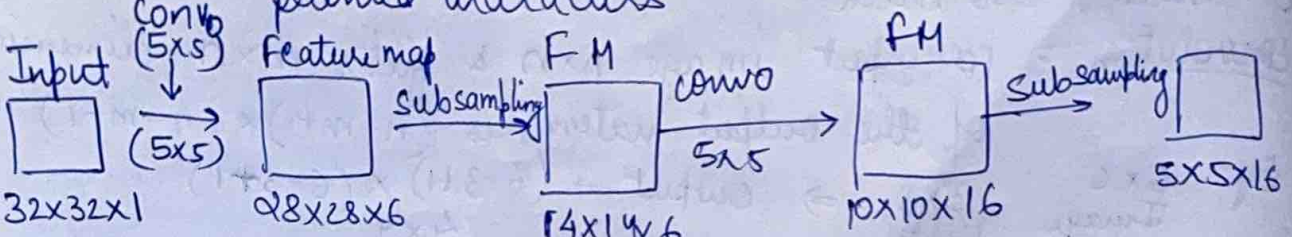
- **Pooling**

1. Image storage
2. Translation invariance

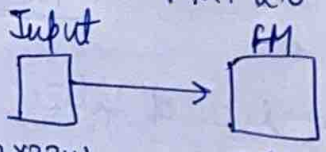
12/11 Popular CNN Architectures

1. LeNet-5
2. AlexNet
3. VGG-16
4. ResNet

LeNet-5 - For recognizing handwritten and machine-printed characters



Output shapes for



$= (32-5+1) \times (32-5+1)$

~~2. Inception~~ 2. AlexNet 3. VGG-16

- Traditional vs Transfer learning

14/11 Normalization

$x \rightarrow$ input, μ_x = mean, σ_x = variance / SD

$$\hat{x}_N = \frac{x - \mu_x}{\sigma_x}$$

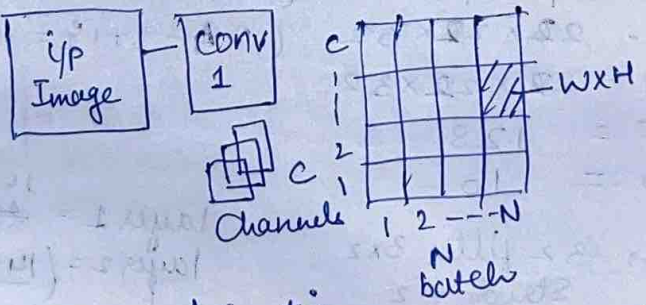
x_{min} & $x_{max} \Rightarrow \hat{x}_N = \frac{x_i - x_{min}}{x_{max} - x_{min}}$

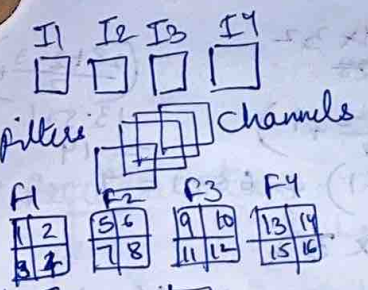
mini-batch ☐☐☐☐ i/p & hidden layers

Batch normalization → Classifier

$x \rightarrow$ mini batch data: μ_B = mean x, σ_B = std of x

$$\hat{x}_B = \frac{x - \mu_B}{\sqrt{\varepsilon + \sigma_B^2}}$$





Reparameterization

→ Learn means & variance of a class through channel

BP

$$\mu_c = \frac{1}{N \times W \times H} \sum_{i=1}^{N} \sum_{j=1}^{W} \sum_{k=1}^{H} x_{iCjk}$$

(x_{iCjk}: i = Batch no., C = Data channel, j = row, k = col = index)

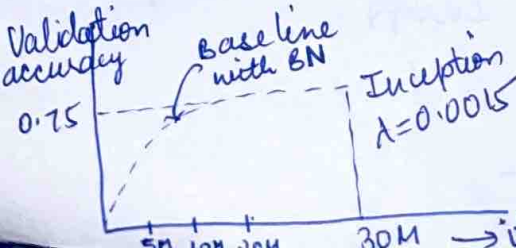
$$\sigma_c^2 = \frac{1}{N \times W \times H} \sum_{i=1}^{N} \sum_{j=1}^{W} \sum_{k=1}^{H} (x_{iCjk} - \mu_C)^2$$

$$\hat{x} = \frac{x - \mu_C}{\sqrt{\sigma_C^2 + \varepsilon}}$$

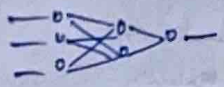
→ γ & β are trainable

$$y_i = \gamma \hat{x}_i + \beta$$

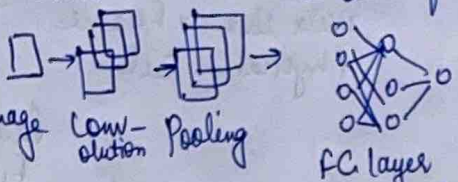
(γ: Std, β: mean)



18/11 RNN DL

ANN 

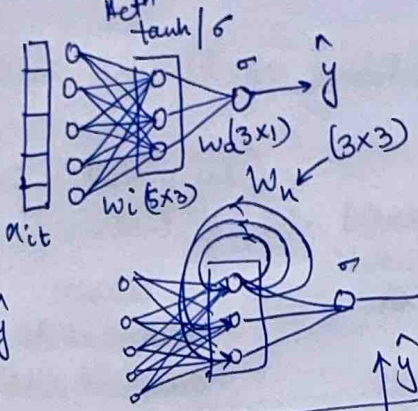
DL → MLP → size of input is fixed
↘ CNN Not suitable for image



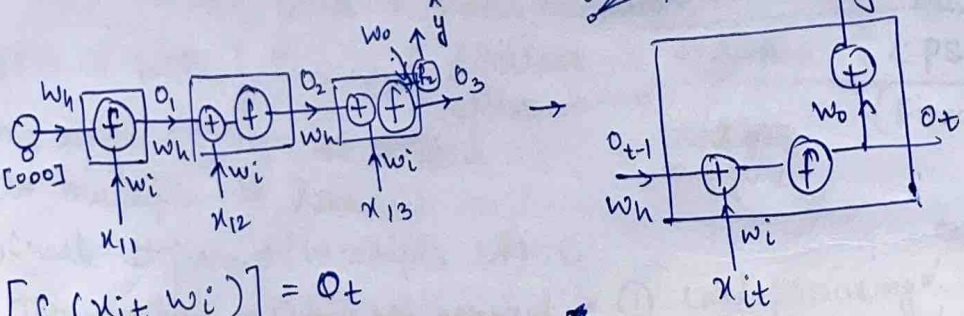
text analysis → size of i/p is not fixed

RNN is used for sequential data. which specifies the timestamps as well.

	x_{11}	x_{12}	x_{13}			y
S_1	v_1	v_2	v_3			1
S_2	v_1	v_2	v_3	v_4	v_5	0
S_3	x_{21}	x_{22}	x_{23}	x_{24}	x_{25}	
⋮						



$f_2([f_1(x_{11} \cdot w_i)]\, w_0) \rightarrow \hat{y}$



$[f_1(x_{it} w_i)] = O_t$

$\hat{y} = \left[f\left[(O_{t-1} \cdot w_h) + w_i x_{it}\right] w_0\right]$

$w_i = w_i - \eta \frac{\partial L}{\partial w_i}$

$w_0 = w_0 - \eta \frac{\partial L}{\partial w_0}$

$w_h = w_h - \eta \frac{\partial L}{\partial w_h}$

Back-Prop-Through Time

$L = -y \log \hat{y} - (1-y) \log(1-\hat{y})$

$O_0 - O_1\ O_2\ O_3$

$\hat{y} \rightarrow (O_3 w_0)$

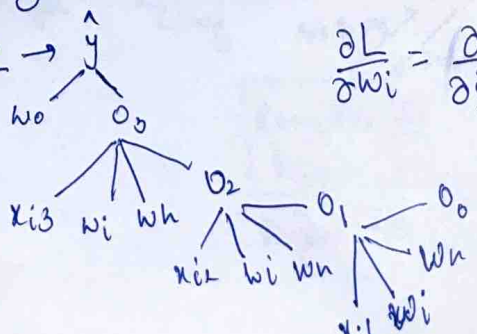
$\frac{\partial L}{\partial w_0} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial w_0}$

$\frac{\partial L}{\partial \hat{y}} = -\left[\frac{y}{\hat{y}} - \frac{1-y}{(1-\hat{y})}\right]$

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

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

$\frac{\partial \sigma(O_3 w_0)}{\partial w_0} = \sigma(O_3 w_0)(1-\sigma(O_3 w_0)) \cdot w_0$

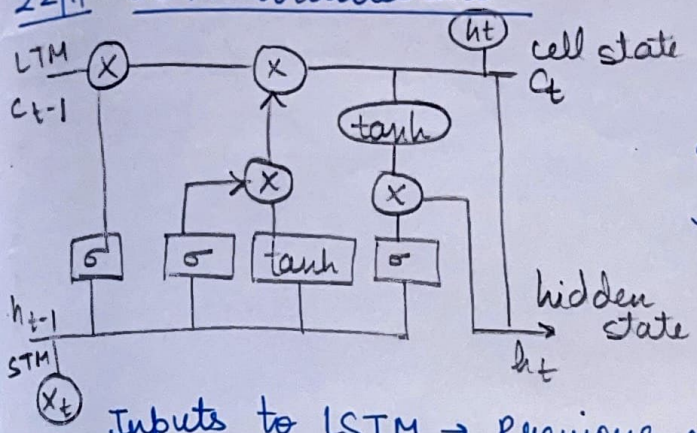


$\frac{\partial L}{\partial w_i} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial O_3} \frac{\partial O_3}{\partial w_i} + \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial O_3} \frac{\partial O_3}{\partial O_2} \frac{\partial O_2}{\partial w_i} +$

$\frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial O_3} \frac{\partial O_3}{\partial O_2} \frac{\partial O_2}{\partial O_1} \frac{\partial O_1}{\partial w_i}$

$$\frac{\partial L}{\partial w_h} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial o_3} \frac{\partial o_3}{\partial w_h} + \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial o_3} \frac{\partial o_3}{\partial o_2} \frac{\partial o_2}{\partial w_h} + \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial o_3} \frac{\partial o_3}{\partial o_2} \frac{\partial o_2}{\partial o_1} \frac{\partial o_1}{\partial w_h}$$

22/11 **LSTM Architecture**



long term memory is cell state

Short term memory is in hidden state

Content – long term content
short term

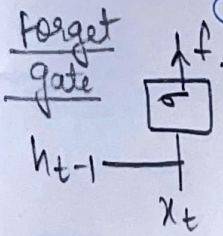
Inputs to LSTM → Previous cell state – C_{t-1}
Previous hidden state – h_{t-1}
i/p at current time index – x_t

Outputs generated by LSTM → current hidden state – h_t
current cell state – C_t

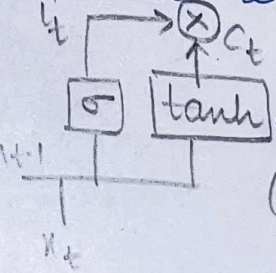
Process of LSTM →1. update cell state i.e. long term content from C_{t-1} to C_t

2. Calculate current hidden state i.e. short term content → h_{t-1} to h_t

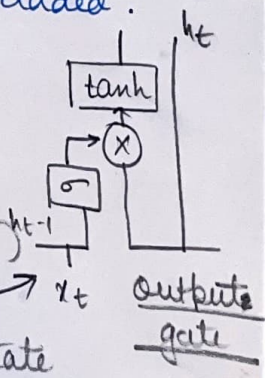
* C_t → Based on current i/p decide what to remove from long term memory & what new to be added.



To remove from long term memory



To add to cell state (update long term memory)

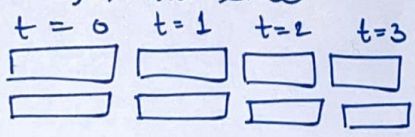


* **Data dimension**

x_t → vector

h_t & C_t → also vector, same size

Data → Sent 1, Sent 2, Sent 3



x_t one word vector

Dimension of x_t & C_t, h_t need not be same

* output of forget gate is f_t

* Other o/p vectors in LSTM are i_t, C_t, O_t ↑ are of same dimension as that of C_t & h_t

* **Operations**

⊗ element wise multiplication

⊕ element wise addition

tanh element wise tanh

Eg: $[1,2,3] \otimes [1,4,7] = [1\times1, 2\times4, 3\times7]$

Eg: $\tanh(1,4,3) = [\tanh(1), \tanh(4), \tanh(3)]$