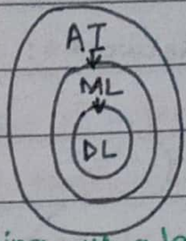
DEEP LEARNING FOR PERCEPTION

Date ________



→ Deep Learning
- Deep → Deep forest
- Learning → Neural Networks → GANS, NLP

→ 3 Types of learning in ML
- Supervised
- Unsupervised
- Reinforcement

Supervised = Learning with a labeled training set

Unsupervised = Discover patterns in unlabeled data

Reinforcement = Learn to act based on feedback / reward

ML vs DL

ML ⇒ Input dataset → Feature extraction → Classification

↳ Small dataset is also acceptable

Feature extraction:
- ↳ Human dependant
- ↳ Difficult
- ↳ Human biasness

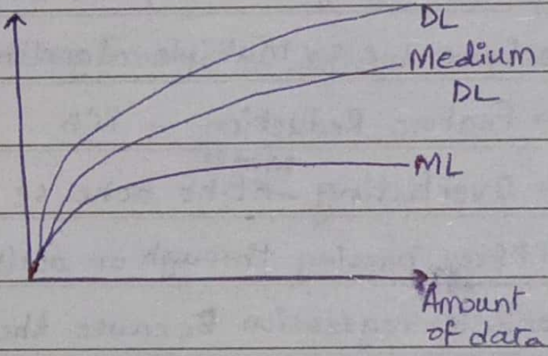
Classification:
- ↳ Depends on FE

DL (Merges FE and Classification) ⇒ Input data → Extracts Features + Classification
- Extracts Features: ↳ Not human dependant
- Classification: → Fine-tuned

↳ Through backpropogation, classification is fine-tuned which improves the accuracy

↳ Drawback requires a huge dataset because it needs to learn on its own.

↳ Why is it so hyped?
- → Processing Power has gotten cheap
- → Big Data
- → Algorithms advancement

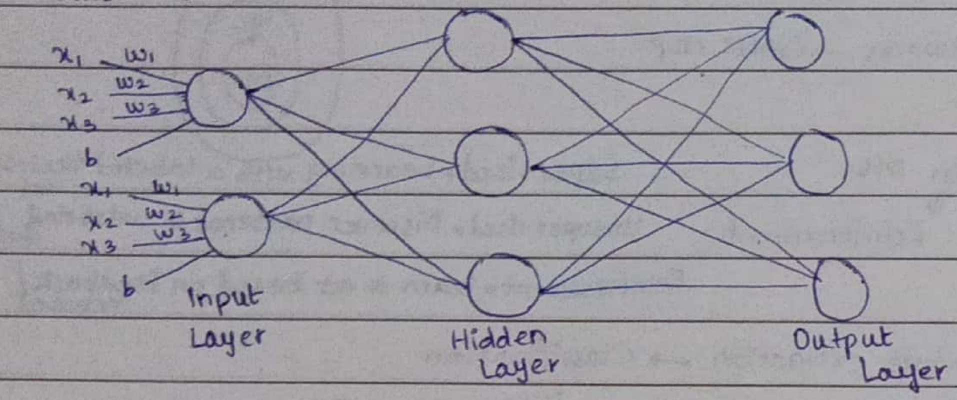


→ The larger the data, the better the accuracy.

Why is DL useful?
- ↳ Learned features are easy to adapt, fast to learn
- ↳ Provides a very flexible, almost universal, learnable framework
- ↳ Manually designed features are often over-specified, incomplete and takes a long time to design and validate

Date ______

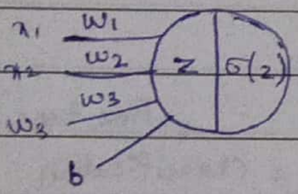
Neural Networks:



This is fully connected network but it is not always necessary.

W = weight

b = bias



$w_1x_1 + w_2x_2 + w_3x_3 + b$ → Output → Activation function

↓ simple linear regression equation

Activation function ↓
→ Adds non-linearity
→ Check krta hai kis value ko forward krna hai
→ Ek range mei laata hai

NN hidden layers <=2

DNN hidden Layers >2

→ Vanishing gradient problem in ML because of which got stuck.

→ Ensemble → Multiple algorithms ko le kr merge kr diya (bagging, boosting)

→ Feature Reduction → PCA

→ Overfitting → Model Bht ache se yaad kr leta hai features

→ After passing through an activation function, linear regression becomes logistic regression because the values are mapped from $(-\alpha, +\alpha)$ to either $(-1,1)$ or $(0,1)$ [RELU].

Forward Pass

1: $z = wx + b$

$a = g(z)$

2: $z = w^l a^{l-1} + b^l$

$a = g(z)$

Backward Pass (Derivative is taken)

1 = $\partial a = g(z)$

2. $\partial z = wa + b$

$\partial a = g(z)$

$$W^l = W^l - \alpha \, dW^l$$

↓ learning rate → kis bhaav se weights update horhe hain

Date

Parameters versus hyperparameters

Weights and bias — learning rate, number of iterations, number of layers, number of hidden units, choice of activation function

→ Parameters that control the weight parameters (hyperparameters)
↳ Should be fixed prior to training the NN

•> Model parameters are estimated from data during the training step

•> Model hyperparameters are manually set earlier and are used to assist estimation model parameter.

0 to 1 → Classification Algorithm (KNN, Naive Bayes) Linear = Continuous

Logistic Regression:- → Discrete

$z = \sum_{i=1}^{n} w_i x_i + b$ $h = \text{Sigmoid}(z)$ → Activation function

$\hookrightarrow \frac{1}{1+e^{-z}}$

Naive Bayes vs Logistic Regression

↳ Generative Classifier → Discriminative Classifier

↳ Learns more features and can regenerate it

↳ tries to find a unique feature to differentiate it.

$\sigma(z) = \frac{1}{1+e^{-z}} = z = -\alpha = \frac{1}{1+e^{\infty}} \sim 0$

$= \frac{1}{1+e^{-z}} = z = \alpha = \frac{1}{1+e^{-\infty}} \sim 1$

$x = [3, 2, 1, 3, 0, 4.19]$	$x = [2, 3, 5, 7, 1]$
$w = [2.5, -5, -1.2, 0.5, 2.0, 0.7]$	$w = [5, 6.1, 7.8, 2.3, 1]$
$b = 0.1$	$b = 0.2$
$\sum_{i=1}^{n=6} w_i x_i + b = 0.733$	$= 84.4$
$P(y=1 \mid x)$	
$P(y=0 \mid x) = 0.3$	

Date

$$P(y=1|x) = \frac{1}{1+e^{-z}}$$

$$P(y=0|x) = 1 - \frac{1}{1+e^{-z}} = \frac{1+e^{-z}-1}{1+e^{-z}} = \frac{e^{-z}}{1+e^{-z}}$$

$L(\hat{y}, y)$ — y: true value, $\hat{y}$: predicted value

Cross entropy, binary

$L_{CE} =$

$$P(y|x) = \hat{y}^{y}(1-\hat{y})^{1-y}$$

$$= \log\left(\hat{y}^{y}(1-\hat{y})^{1-y}\right)$$

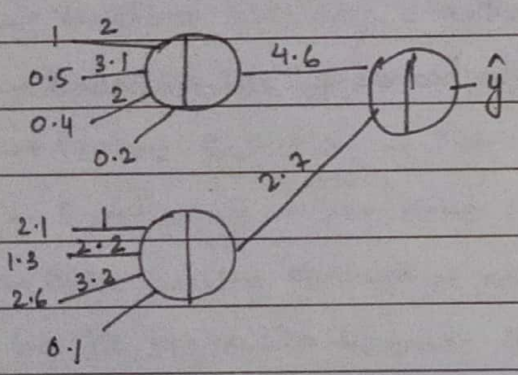
$$= \log \hat{y}^{y} + \log(1-\hat{y})^{1-y}$$

$$y \log \hat{y} + (1-y)\log(1-\hat{y}) \rightarrow \text{Log Likelihood}$$

$$y \log \sigma(z) + (1-y)\log(1-\sigma(z)) \qquad \sigma(z) = a$$

$$L_{CE} = y\log(a) + (1-y)\log a$$

Types of Logistic Regression
- Binary (2 classes)
- Multinomial (more than 2)
- Ordinal (More than 2 classes + order matters)



$y = 0.98$

1: $z_1 = 1(2) + 0.5(3.1) + 0.4(2) + 0.2 = 4.55$ $\qquad L_{CE} = -0.00102$

$a = 1/1+e^{-4.55} = 0.9895$

$z_2 = 2.1(1) + 1.3(2.2) + 2.6(3.2) + 0.1 = 13.38$

$a = 0.999998$

2:

$z_1 = 0.9895(4.6) + 0.999998(2.7) = 7.25$

$a = 0.99929$

$\hat{y} = 0.99929$

Date

$L_{CE}(\hat{y}, y) = -\log P(y|x)$

$-\left[y\log(\hat{y}) + (1-y)\log(1-\hat{y})\right]$ General Cross Entropy

Binary Cross Entropy (2 classes)

GRADIENT

$a(1-a).\,x_1$

$$\frac{\partial L}{\partial w_1} = \frac{\partial L}{\partial a} \cdot \frac{\partial a}{\partial z} \cdot \frac{\partial z}{\partial w_1}$$

$$\frac{\partial}{\partial a}\left[-y\log(a) - (1-y)\log(1-a)\right]$$

derivative of sigmoid $= a(1-a)$

$$\frac{-y}{a} - \frac{1(-1)}{1-a}(1-y) = \frac{-y}{a} + \frac{1-y}{1-a}$$

$z = w_1x_1 + w_2x_2 + b$

$a = \hat{y} = \sigma(z)$

$$\therefore \left[\frac{-y}{a} + \frac{1-y}{1-a}\right] \cdot a(1-a) \cdot x_1$$

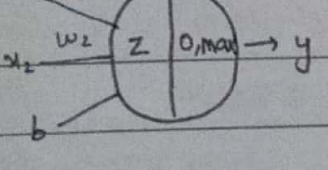
$$\frac{-y(1-a) + a(1-y)}{\cancel{a(1-a)}} \cdot \cancel{a(1-a)} \cdot x_1$$

$$(-y + \cancel{ay} + a - \cancel{ay}) \cdot x_1 = (a-y) \cdot x_1$$

$$\frac{\partial L}{\partial w_2} = \frac{\partial L}{\partial a} \cdot \frac{\partial a}{\partial z} \cdot \frac{\partial z}{\partial w_2}$$

$$(a-y) \cdot x_2$$

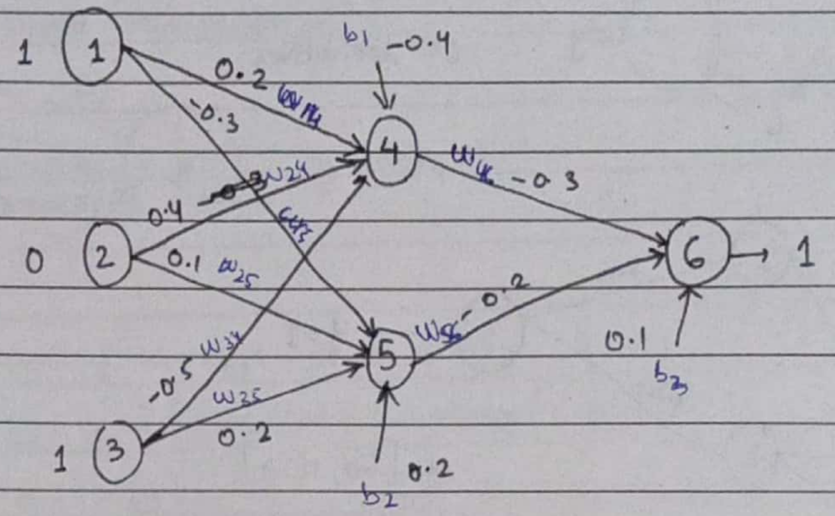
$$\frac{\partial L}{\partial b} = a - y$$



AND Gate (0,0,-1)

OR Gate (1,1,0)

Date



$\eta = 0.9$

$z_4 = -0.7$ $\qquad$ $z_5 = 0.1$

$a_4 = 0.332$ $\qquad$ $a_5 = 0.525$

$z_6 = \cancel{0.0546}$ -0.10403

$a_6 = \cancel{0.524}$ 0.4734

$L = \frac{1}{2}(\text{target} - \overset{\text{predicted}}{\text{output}})^2$ − MSE

$= \frac{1}{2}(1 - 0.474)^2 = 0.138$

$\frac{\partial L}{\partial w_{46}} = \frac{\partial L}{\partial a} \cdot \frac{\partial a}{\partial z} \cdot \frac{\partial z}{\partial w_{46}}$

$= (a - \text{target}) \cdot (a)(1-a) \cdot a_4$

$= (0.474 - 1)(0.474)(1 - 0.474) \cdot (0.332)$

$= -0.0435399$

$w_{46}^{new} = w_{46}^{old} - \eta \frac{\partial L}{\partial w_{46}}$

$= -0.2608 \approx -0.261$

$\frac{\partial L}{\partial w_{56}} = -0.068885$

$w_{56}^{new} = w_{56}^{old} - \eta \frac{\partial L}{\partial w_{56}}$

$\frac{\partial L}{\partial b_3} = -0.1311$

$= \cancel{-0.13499}$ $-0.138035 \simeq -0.138$

$b_3^{new} = b_3^{old} - \eta \frac{\partial L}{\partial b_3} = 0.21799 \simeq 0.218$

Date

$$\frac{\partial L}{\partial w_{14}} = \frac{\partial L}{\partial a_6} \cdot \frac{\partial a_6}{\partial z_6} \cdot \frac{\partial z_6}{\partial a_4} \cdot \frac{\partial a_4}{\partial z_4} \cdot \frac{\partial z_4}{\partial w_{14}}$$

$$= \frac{\partial L}{\partial b_3} \cdot \frac{\partial z_6}{\partial a_4} \cdot \frac{\partial a_4}{\partial z_4} \cdot \frac{\partial z_4}{\partial w_{14}}$$

$$= (a_6 - \text{target})(a_6)(1-a_6)(w_{46}) \cdot (a_4)(1-a_4)(x_1)$$

$$\frac{\partial L}{\partial w_{15}} = \frac{\partial L}{\partial a_6} \cdot \frac{\partial a_6}{\partial z_6} \cdot \frac{\partial z_6}{\partial a_5} \cdot \frac{\partial a_5}{\partial z_5} \cdot \frac{\partial z_5}{\partial w_{15}}$$

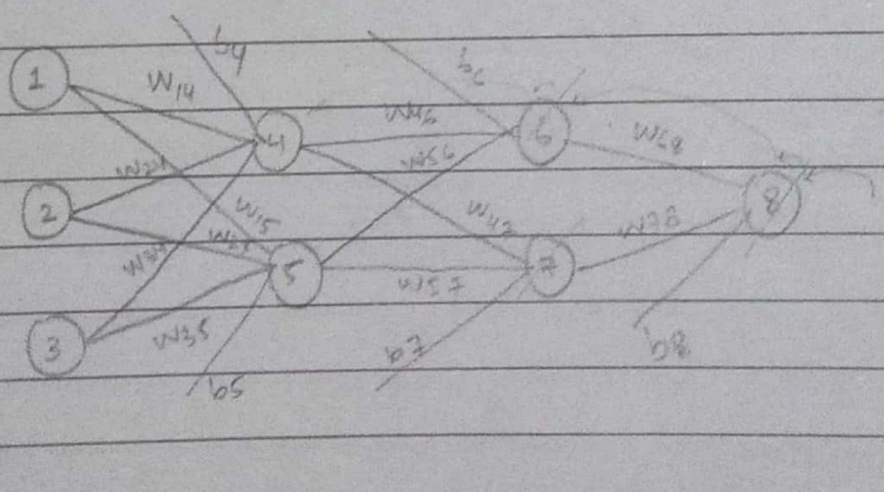
$$= (a_6 - \text{target})(a_6)(1-a_6)(w_{56}) \cdot (a_5)(1-a_5)(x_1)$$

$$\frac{\partial L}{\partial w_{24}} = (a_6 - \text{target})(a_6)(1-a_6)(w_{46})(a_4)(1-a_4)(x_2)$$

$$\frac{\partial L}{\partial w_{34}} = (a_6 - \text{target})(a_6)(1-a_6)(w_{46})(a_4)(1-a_4)(x_3)$$

$$\frac{\partial L}{\partial_{25}} = (a_6 - \text{target})(a_6)(1-a_6)(w_{56})(a_5)(1-a_5)(x_2)$$

$$\frac{\partial L}{\partial_{35}} = (a_6 - \text{target})(a_6)(1-a_6)(w_{56})(a_5)(1-a_5)(x_3)$$



Date

Sigmoid $= \frac{1}{1+e^{-z}}$

tanh $\cdot \frac{1-e^{-z}}{1+e^{-z}}$

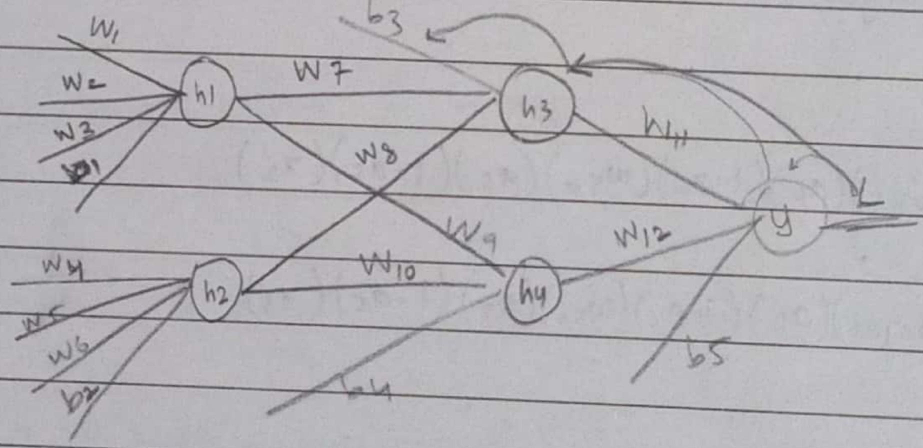
Relu $\max(0, z)$

Softmax $\frac{e^{z_i}}{\sum_{j=1}^{n} e^{z_j}}$

$$\frac{\partial L}{\partial w_{14}} = \frac{\partial L}{\partial a_8} \cdot \frac{\partial a_8}{\partial z_8} \cdot \frac{\partial z_8}{\partial a_6} \cdot \frac{\partial a_6}{\partial z_6} \cdot \frac{\partial z_6}{\partial a_4} \cdot \frac{\partial a_4}{\partial z_4} \cdot \frac{\partial z_4}{\partial w_{14}}$$

$$L = \frac{1}{2}(y - a_8)^2$$

$$\frac{\partial L}{\partial w_{14}} = -(y-a_8) \cdot (a_8)(1-a_8) \cdot w_{68} \cdot (a_6)(1-a_6) \cdot w_{46} \cdot (a_4)(1-a_4) \cdot x_1$$



$$\frac{\partial L}{\partial h_3}$$

Code,
Regularization, ← DLP → Ruhal
Dropout

Adv, SMD → Isbah.

ICC → Both

FYP → Code → Ruhal

Paper → Isbah

Dance → Mahnoor

Date

Regularization

→ To avoid overfitting — training result good, testing result fail

→ Example L1, L2, Early stopping

Regularization (Prevent overfitting)

Optimization (Prevent underfitting) — training result bad

True Loss → Generalization error

Empirical Loss → Test error (val error)

L1: regularization coefficient

Cost = $(\text{loss}) + \lambda \sum_{i=1}^{n} \|w\| \quad \cancel{+ \sum_{i=1}^{n} \|w\|^2}$

L2

$+ \lambda \sum_{i=1}^{n} \|W\|^2$

Drop out / Gradient Descent

Scanned with CamScanner

Date

1 +

2 -

4 +

5 +

6 -

7 -

leaving '1' out

$S_p = \{4, 5\}$ $\mu_p = 4.5$

$S_n = \{2, 6, 7\}$ $\mu_n = 5$

1 + correct

leaving '2' out

$S_p = \{1, 4, 5\}$ $\mu_p = 3.33$

$S_n = \{6, 7\}$ $\mu_n = 6.5$

2 + wrong

leaving '4' out

$S_p = \{1, 5\}$ $\mu_p = 3$

$S_n = \{2, 6, 7\}$ $\mu_n = 5$

4 + correct

leaving '5' out

$S_p = \{1, 4\}$ $\mu_p = 2.5$

$S_n = \{2, 6, 7\}$ $\mu_n = 5$

5 - ~~correct~~ wrong

leaving '6' out

$S_p = \{1, 4, 5\}$ μ_p 3.33

$S_n = \{2, 7\}$ $\mu_n = 4.5$

6 - correct

leaving '7' out

$S_p = \{1, 4, 5\}$ $\mu_p = 3.33$

$S_n = \{2, 6\}$ $\mu_n = 4$

7 - correct

Date

CV

1

train

A	1	+
B	2	-
C	4	+
D	5	+

test

E	6	-
F	7	-

$S_p = \{1, 4, 5\}$ $\mu_p = 3.33$

$S_n = \{2\}$ $\mu_n = 2$

E = + wrong

F = + wrong

2

A	1	+
B	2	-
E	6	-
F	7	-
C	4	+
D	5	+

$S_p \{1\}$ $\mu_p = 1$

$S_n \{2, 6, 7\}$ $\mu_n = 5$

C = 4 - wrong

D = 5 - wrong

3

C	4	+
D	5	+
E	6	-
F	7	-
A	1	+
B	2	-

$S_p = \{4, 5\}$ 4.5

$S_n = \{6, 7\}$ 6.5

1 + correct

2 + wrong

Date

Regularization:-

→ A technique which makes slight modifications to the learning algorithm such that the model generalizes better.

→ To avoid overfitting

→ Penalizes the coefficients → In ML

→ Penalizes the weight matrices of the nodes → In DL

→ If regularization coefficient is so high that some of the weight matrices are nearly equal to zero → slight underfitting of the training data

→ Need to optimize the value of regularization coefficient

↳ Data Augmentation

↳ Early Stopping

↳ L1

↳ L2

L1 Regularization

→ Prevent overfitting of a model by adding a penalty term to the model's cost function,

Lasso $L1 = \lambda * \Sigma |w|$

$|w|$ → Model's weights or coefficients

λ → Regularization Strength (hyper parameter)

$\Sigma |w|$ → Represents the sum of absolute values of all weights

Higher λ → Reduce model's predictive power

lower λ → Increase the risk of overfitting

L2 Regularization

→ Prevent overfitting of a model by adding a penalty term to the model's cost function

Ridge ← $L2 = \lambda * \Sigma (w^2)$ → Sum of the squares of all weights

→ L2 differs from L1. L2 encourages the model's weights to be small and

Date

close to zero but doesnot necessarily set them exactly to zero, as L1 regularization does. L2 regularization is often used when all the features or variables are considered relevant, and a small weight is preferable to no weight for preventing overfitting.

DROPOUT

- NN overfits
- Optimal solution is to use a Bayesian framework. Infeasible when networks are big
- Dropout suggests that each unit should work with a random sample of other units.
- ChatGPT gave an example: "Think of it like a team of workers where some workers take time off during each workday. Each worker learns to do all the tasks necessary for the job, so if any one worker takes a day off, the team can still function effectively.
- At training (each iteration): Each unit is retained with a probability p.
- At test: The network is used as a whole. The weights are scaled down by a factor of p
- Dropout trains 2^n networks (n-number of units)

OR

- At training: Weights are scaled up by a factor of $1/p$
- At test: No scaling applied
  ↳ This method is used in TF.

$$r_i^{(l)} \sim \text{Bernoulli}(p)$$

$$\tilde{y}^{(l)} = r^{(l)} * y^{(l)}$$

$$z_i^{(l+1)} = w_i^{(l+1)} \tilde{y}^{(l)} + b_i^{(l+1)}$$

$$y_i^{(l+1)} = f\left(z_i^{(l+1)}\right)$$

Date

The effect of dropout on learned features:

- Without dropout, units tend to compensate for mistakes of other units
- This leads to overfitting, since the co-adaptations do not generalize to unseen data
- Dropout prevents co-adaptations by making the presence of other hidden units unreliable.

Weight Decay

- Limiting the growth of the weights in the network.

$$L_{new}(w) = L_{old}(w) + \frac{1}{2}\lambda \|w\|_2^2$$

→ Dropout has more advantages over weight decay.

- Dropout is scale-free
- Dropout is invariant to parameter scaling.

→ Dropout is a very good and fast regularization method.

→ Dropout is a bit slow to train (2-3 times slower than without dropout)

→