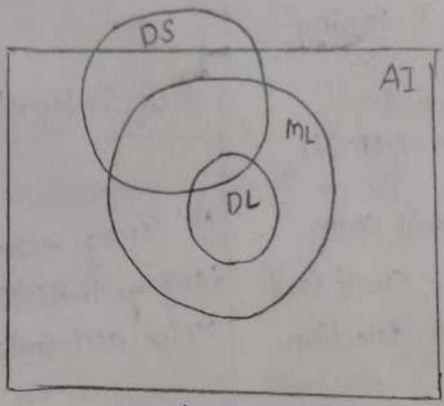
M.Kavya Sri
22A71A3151
CAI-A

①

# Deep Learning
## Assignment-1

1. Differentiate AI vs ML vs DL vs DS




VIEN DIAGRAM.

| Artificial Intelligence | Machine Learning | Deeplearning | Data science. |
|---|---|---|---|
| → Creating machines that can perform tasks requiring human intelligence | A subset of AI where machines learns from data without Explict programming | A subset of ML that uses deep neural networks to process data. | A multidisciplinary field that focuces on extracting insights from data. |
| → Broad (includes ML, DL, rule based System & more) | Narrower than AI, focus on learning from data | Very narrow specifically uses neural networks | Broad (includes AI, ML, statics & big data. |
| → Rule based Systems, Expert systems, Search algorithms | Supervised and unsupervised & reinforcement learning. | Convolutional Neural Networks, Recurrent Neural Networks Transformers. | Statistical analysis, data mining, visualization, ML and DL |
| → Data dependency is Low-Medium | → Data dependency is high (requries lot of data to learn) | Data dependency is very high (requries large labelled dataset) | Data dependency is High (works with structured & un structured data) |
| → Low to High computational power | → Medium to high computational power | very high computation power | Medium to high computation power. |

POCO M6 PRO 5G 21/02/2025 21:00

| | | | |
|---|---|---|---|
| May requried predefined rules. | Requries human intervention for feature selection & tuning | ~~Minmal~~ Minimal intervention, learns features automatically | Requries human expertise in data preprocessing & analysis. |
| Interpretability can be Explainable | Some what Interpretable | Low Interpretability | Medium interpretability |
| Ex: Chatbots, Self driving cars, Recommendation systems | Ex: Email spam filters, Credit card fraud detection | Ex: Image recognisation, language translation, Voice assistants | Predictive analysis, customer segmentation, market trend analysis. |

2. Explain principal component Analysis with an Example using Decomposition method.

Principal Component Analysis [PCA]

PCA technique was introduced by mathematician Karl pearson in 1991 it works on the condition that why the data in a bipolar dimensional space is mapped to data in a lower dimensional space the variance of data in the lower dimensional space should be maximum.

→ PCA is a statistical procedure that uses an orthogonal Transformation that converts a set of correlated variables, to a set of Uncorrelated Variables.

→ PCA is an unsupervised ML Algorithm technique used to examine the inter relations among a set of variables.

→ Main goal of PCA is to reduce the dimensionality of a dataset by preserving the most important patterns or relationship for Example given the following dataset use PCA to reduce dimensions from 2 to 1

Data set:

| feature | Ex1 | Ex2 | Ex3 | Ex4 |
|---|---|---|---|---|
| x | 4 | 8 | 13 | 7 |
| y | 11 | 4 | 5 | 14 |

Step 1: The Given Dataset, No. of features ⇒ $n = 2$

No. of samples $N = 4$

Step 2: competition of mean of variable, $\bar{x} = 8$

$\bar{y} = 8.5$

Step 3: competition of covariance matrix,

$(x,x)\ (x,y)\ (y,x)\ (y,y)$

(i) covariance of $(x,x) = \frac{1}{N-1}\sum_{K=1}^{N}(x_i - \bar{x})^2$

$= \frac{1}{4-1}[4-8]^2 + [8-8]^2 + [13-8]^2 + [7-8]^2$

$= \frac{1}{3}[16+0+25+1]$

$= \frac{1}{3}(42)$

$= \underline{14}$

(ii) covariance of $(x,y) = \frac{1}{N-1}\sum_{K=1}^{N}(x_{ik} - \bar{x}_i)(x_{jk} - \bar{x}_j)$

$= \frac{1}{4-1}[(4-8)(11-8.5) + (8-8)(4-8.5) + (13-8)(5-8.5) + (7-8)(14-8.5)]$

$= \frac{1}{3}[-10 + 0 + (-17.50) + (-5.50)]$

$= \frac{1}{3}[-10 - 17.5 - 5.5] = -11$

POCO M6 PRO 5G 21/02/2025 21:01

(iii) covariance $(y,x) = -11$

(iv) covariance $(y,y) = \frac{1}{N-1}\sum_{K=1}^{N}(y_i-\bar{y})^2$

$$= \frac{1}{4-1}[11-8.5]^2+[4-8.5]^2+[5-8.5]^2+[14-8.5]^2$$

$$= \frac{1}{3}[69]$$

$$= 23$$

Covariance Matrix :-

$$S = \begin{bmatrix} cov(x,x) & cov(x,y) \\ cov(y,x) & cov(y,y) \end{bmatrix} = \begin{bmatrix} 14 & -11 \\ -11 & 23 \end{bmatrix}$$

Step 4 :- TO FIND EIGEN values and EIGEN vectors and Normalize

Eigen Value.

$$\begin{bmatrix} 14-\lambda & -11 \\ -11 & 23-\lambda \end{bmatrix} = (14-\lambda)(23-\lambda)-121=0$$

$$= 322-14\lambda-23\lambda+\lambda^2-121=0$$

$$= \lambda^2-37\lambda+201=0$$

$$\lambda_1 = 30.3849 \quad \lambda_2 = 6.6151$$

$\lambda_1 = 30.3849$

$$\begin{bmatrix} 14-30.3849 & -11 \\ -11 & 23-30.3849 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} = 0 \quad X$$

$$\begin{bmatrix} -16.3849 & -11 \\ -11 & -7.3849 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} = 0 \quad X$$

$$-16.3849x-11y=0$$

POCO M6 PRO 5G 21/02/2025 21:01

$\lambda_1 = 30.3849$

$$\begin{bmatrix} 14-\lambda & -11 \\ -11 & 23-\lambda \end{bmatrix}\begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = 0$$

$(14-\lambda)v_1 - 11v_2 = 0$

$-11v_1 + (23-\lambda)v_2 = 0$

$(14-\lambda)v_1 = 11v_2$

$$\frac{v_1}{11} = \frac{v_2}{14-\lambda}$$

$$\frac{v_1}{11} = \frac{v_2}{14-30.3849}$$

$$\begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} 11 \\ -16.3849 \end{bmatrix}$$

Normalize vector,

$$e_1 = \begin{bmatrix} \frac{11}{\sqrt{(11)^2+(-16.3849)^2}} \\ \frac{-16.3849}{\sqrt{(11)^2+(-16.3849)^2}} \end{bmatrix}$$

$$e_1 = \begin{bmatrix} 0.5574 \\ -0.8303 \end{bmatrix}$$

$\lambda_2 = 6.6151$

$$\frac{v_1}{11} = \frac{v_2}{14-6.6151}$$

$$\frac{v_1}{11} = \frac{v_2}{8.6151}$$

$$\begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} 11 \\ 8.6151 \end{bmatrix}$$

$$e_2 = \begin{bmatrix} \frac{11}{\sqrt{(11)^2+(8.6151)^2}} \\ \frac{8.6151}{\sqrt{(11)^2+(8.6151)^2}} \end{bmatrix}$$

$$e_2 = \begin{bmatrix} 0.8303 \\ 0.3574 \end{bmatrix}$$

POCO M6 PRO 5G 21/02/2025 21:01

Step 5: The New dataset.

| First principle component | $P_{11}$ | $P_{12}$ | $P_{13}$ | $P_{14}$ |
|---|---|---|---|---|
| | -4.3052 | 3.7361 | 5.6923 | -5.1235 |

we find $P_{11}$

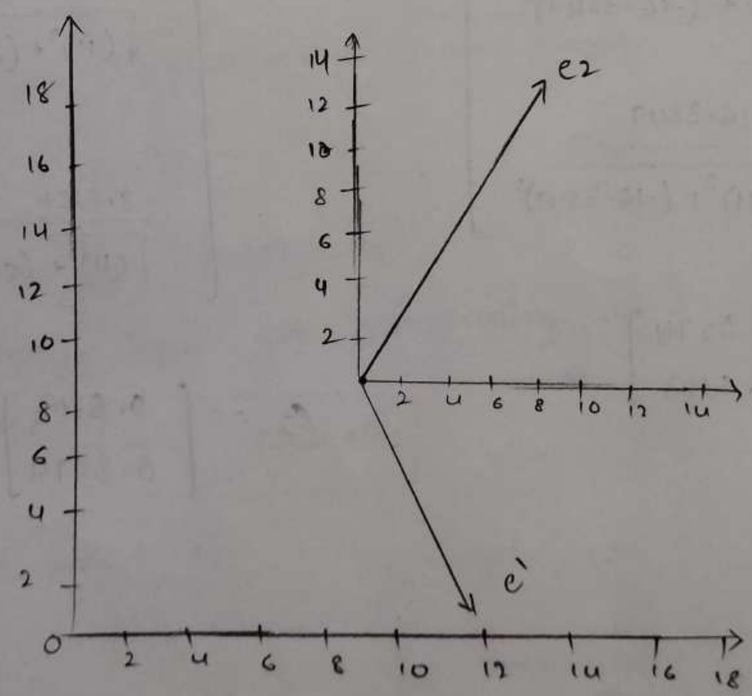
$$P_{11} = e_1^T \begin{bmatrix} 4 - 8 \\ 11 - 8.5 \end{bmatrix} = \begin{bmatrix} 0.5574 & -0.8303 \end{bmatrix} \times \begin{bmatrix} -4 \\ 2.5 \end{bmatrix} = -4.3052$$

$$P_{12} = e_1^T \begin{bmatrix} 8 - 8 \\ 4 - 8.5 \end{bmatrix} = \begin{bmatrix} 0.5574 & -0.8303 \end{bmatrix} \times \begin{bmatrix} 0 \\ -4.5 \end{bmatrix} = 3.7361$$

$$P_{13} = e_1^T \begin{bmatrix} 13 - 8 \\ 5 - 8.5 \end{bmatrix} = \begin{bmatrix} 0.5574 & -0.8303 \end{bmatrix} \times \begin{bmatrix} 5 \\ -3.5 \end{bmatrix} = 5.6928$$

$$P_{14} = e_1^T \begin{bmatrix} 7 - 8 \\ 14 - 8.5 \end{bmatrix} = \cancel{-5.1235} \begin{bmatrix} 0.5574 & -0.8303 \end{bmatrix} \times \begin{bmatrix} -1 \\ +6.5 \end{bmatrix} = -5.1235$$

Step 6: Graphical Representation.



30 Explain forward and backward propagation in Artificial Neural Network (ANN)?

1. **Forward propagation:**

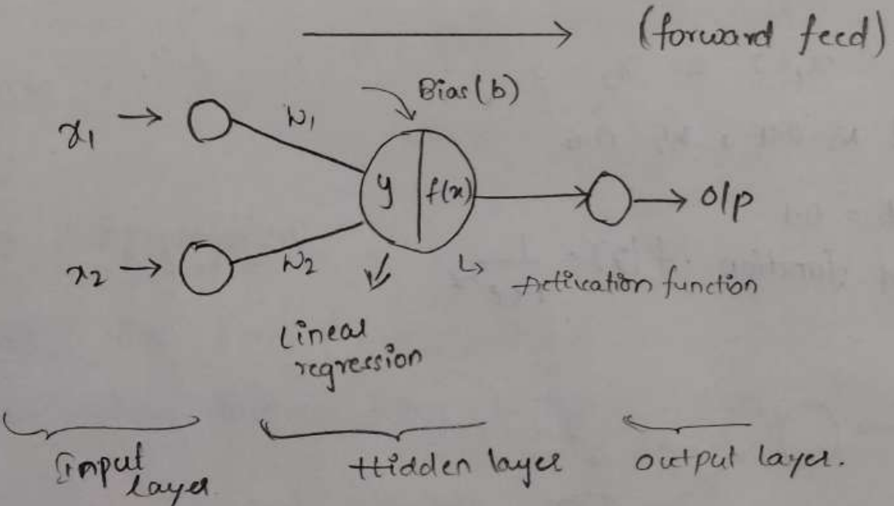
Forward Propagation is the process where input data is passed through the network layer by layer to generate output. It involves.

→ Multiplication of Input with weights.

→ Adding Bias.

→ Applying an activation function.

→ passing the result to the next layer.




**Steps of forward propagation:**

**Input layer:** Take the input Values

**weighted sum:** Multiple inputs with corresponding weights and add bias $\boxed{y = \sum_{i=1}^{n} x_i w_i + b}$

$$y = (w_1x_1 + w_2x_2 + w_3x_3 - - - - + w_nx_n) + b$$

3. Activation function: Apply an activation function like RELU, Sigmoid, Tanh etc. to introduce non-linearity for sigmoid Activation function,

$$\boxed{Z = f(y) = \frac{1}{1+e^{-y}}}$$

4. Output layer:

The final layer computes the output.

Example for forward propogation:

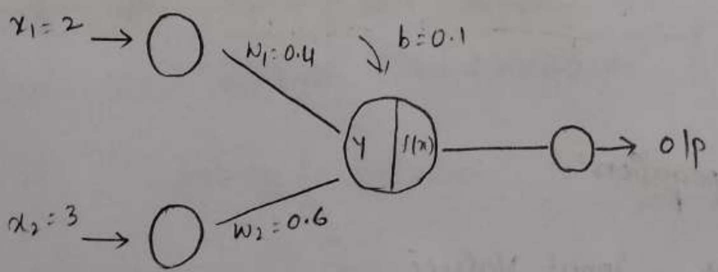
Consider,

input $x_1 = 2$ & $x_2 = 3$

weight $w_1 = 0.4$ , $w_2 = 0.6$

Bias $b = 0.1$

sigmoid function $\cdot f(z) = \frac{1}{1+e^{-z}}$




ie $y = w_1 x_1 + w_2 x_2 + b$

$= (0.4)(2) + (0.6)(3) + 0.1$

$$\boxed{y = 2.7}$$

Sigmoid function,

$$f(y) = \frac{1}{1+e^{-y}} \Rightarrow \frac{1}{1+e^{-2.7}}$$

POCO M6 PRO 5G 21/02/2025 21:01

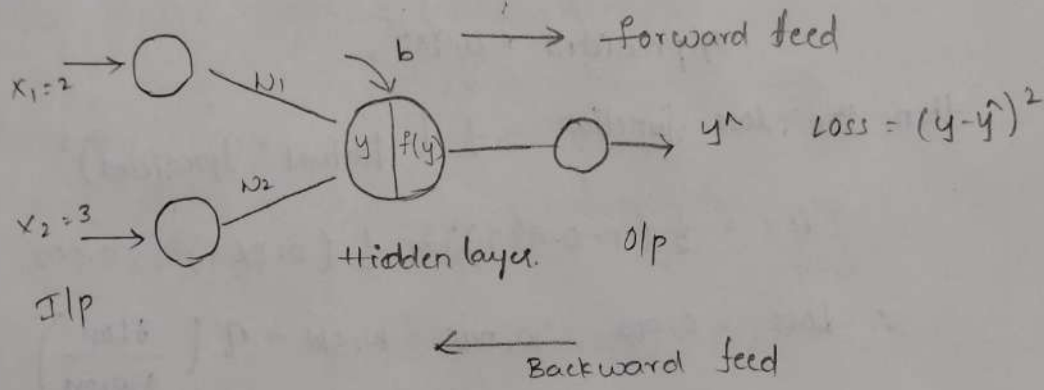
(5)

$$f(y) = 0.937 \quad \text{where} \begin{array}{l} \geq 0.5 \approx 1 \\ < 0.5 \approx 0 \end{array}$$

$\therefore$ Output of forward propagation is $0.937 \approx 1$

**Backward propagation:**

Backward propagation is the process of adjusting the weights to minimize the Error using Gradient Descent




**Steps for Back propagation:**

1. calculate the Error (Loss function):

Use Mean Squared Error (MSE) or Cross Entropy Loss

$$Loss = \frac{1}{2}(y_{actual} - y_{predicted})^2$$

2. compute gradient:

- Differentiate loss function with respect to weights using chain rule and Update the weights using Gradient Descent

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

ie, $$w_{new} = w_{old} - \eta \frac{\partial Loss}{\partial w_{old}}$$

where $\eta \rightarrow$ learning rate, $\frac{\partial L}{\partial w} \rightarrow$ slope (or) Gradient Descent

POCO M6 PRO 5G 21/02/2025 21:01

3. Adjust Weights and Biases:

- Repeat untill Loss is minimized

Example:

from the forward propogation, the actual Output is

$\therefore y_{actual} = 1$

then the predicted value $= 0.937$

$\therefore y_{predicted} = 0.937$,

then the loss function $= \frac{1}{2}(y_{actual} - y_{predicted})^2$

ie $= \frac{1}{2}(1-0.937)^2 \Rightarrow \frac{1}{2}(0.063)^2 \Rightarrow 0.002$

$\therefore$ Loss $= 0.002$, $w_1 new = w_1 old - \eta\left(\frac{\partial Loss}{\partial w_1 old}\right)$

then, $\frac{\partial Loss}{\partial w_1 old} = \frac{\partial Loss}{\partial y_{predicted}} \times \frac{\partial y_{predicted}}{\partial z} \times \frac{\partial z}{\partial w_1}$

ie $\frac{\partial Loss}{\partial y_{predicted}} = y_{predicted} - y_{actual}$

$= 0.937 - 1 \Rightarrow -0.063$

Derivative of output wrt Activation function (sigmoid)

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

$f'(z) = f(z) \times (1 - f(z))$

$= 0.937 \times (1 - 0.937)$

$= 0.059$

POCO M6 PRO 5G 21/02/2025 21:02

wrt weight $w_1$,

$$\frac{\partial z}{\partial w_1} = x_1 = 2 \quad , \quad \frac{\partial z}{\partial w_2} = x_2 = 3$$

then,

$$\frac{\partial Loss}{\partial w_1} = \frac{\partial Loss}{\partial y_{predicted}} \times \frac{\partial y_{predicted}}{\partial z} \times \frac{\partial z}{\partial w_1}$$

$$(-0.063) * (0.059) * (2)$$

$$= -0.007434$$

Similarly,

$$\frac{\partial Loss}{\partial w_2} = \frac{\partial Loss}{\partial y_{predicted}} \times \frac{\partial y_{predicted}}{\partial z} \times \frac{\partial z}{\partial w_2}$$

$$= (-0.063) * (0.059) * (3)$$

$$= -0.011151$$

then, Here $\eta = 0.1$

$$w_1 \text{ new} = w_1 \text{ old} - \eta \times (-0.007434)$$

$$= 0.4 - 0.1 (-0.007434)$$

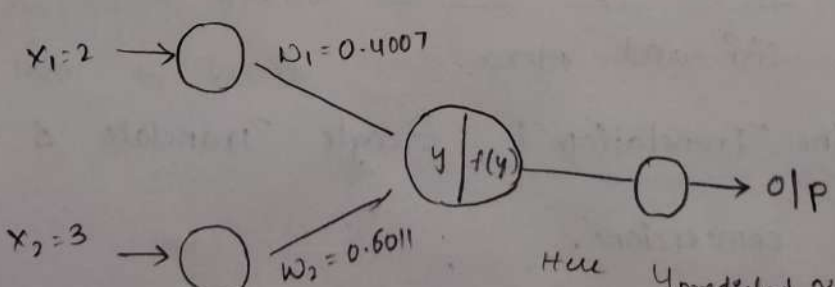
$$\boxed{w_1 \text{ new} = 0.40074}$$

$$w_2 \text{ new} = w_2 \text{ old} - 0.1 * (-0.011151)$$

$$= 0.4 - 0.1 (-0.011151)$$

$$\boxed{w_2 \text{ new} = 0.6011}$$

then,




Here $y_{predicted} \approx y_{actual}$

Here y predicted and y actual Values would be Similar (or) Same after the Updation of weights, where the Error rate (or) the loss function Value minimizes (or) reduces.

4Q Write a Short note on,

A) Applications of Deep Learning.

B) Activation function

c) optimizers.

D) Loss functions.

1 Applications of Deep learning:

Deep Learning (DL) is widely used in various fields due to its ability to process large amount of data and complex patterns.

1. Compute Vision:

A) Image classification: Identifies Objects in Images (Eg cat or Dog)

B) Object Detection: Detects & labels multiple objects in an image (Eg self-driving cars, detecting pedestrains)

c) Facial Recognition: used in Security Systems and social Media. (Eg: Facebook auto-tagging)

2. Natural Language processing (NLP):

- Chatbots & Virtual Assistants: AI-powered bots like Chat GPT, Siri and Alexa.

- Machine Translation: Google Translate & DeepL for Language conversions.

POCO M6 PRO 5G 21/02/2025 21:02

Example:

• Sentiment analysis: Analyzing customer feedback, reviews and Social Media Sentiment.

3. Healthcare:

• Disease Diagnosis: AI Models detect diseases like cancer from X-rays and MRI's.

• Drug Discovery: predicts chemical interactions for new medicine.

• personalized Treatment: AI recommend treatments based on patient history.

4. Finance:

• Fraud Detection: identifies fradulent transactions in banking.

• Stock market prediction: AI powered trading bots analyze stock trends.

5. Speech Recognition:

• Voice Assistants: Alexa, siri, Google Assistant convert speech to text

• Automated ~~Transcri~~ Transcription: AI tools like otter.ai transcribe meetings.

6. Recommendation systems:

Netflix, youtube, Amazon: suggests movies, videos & products. based on user behaviour.

(B) Activation Functions:

Activation function introduce non-linearity in neural networks, allowing them to learn complex relationships.

POCO M6 PRO 5G 21/02/2025 21:03

## Types of Activation functions:

1. Sigmoid function: (Good for binary classification)

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

Output range $= (0,1)$

**Example**

if $z=2$ then

$$f(2) = \frac{1}{1+e^{-2}} \approx 0.88$$

**problem**: Causes Vanishing gradient in deep networks

2. **RELU** (Rectified Linear Unit)

→ It mostly used Activation function

$$\boxed{f(z) = \max(0,z)}$$

Output range $= [0, \infty)$

**Example** $f(3) = 3$

$f(-2) = 0$

Advantage: faster learning & avoids vanishing gradients

problem: Dying RELU (neurons stop updating for negative inputs)

3. **Tanh (Hyperbolic Tangent):**

It is better than sigmoid

$$f(x) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}}$$

Output range $= (-1, 1)$

Example:

$f(2) = 0.96 \qquad f(-2) = -0.96$

Advantage: Helps in zero-centered learning

4. Softmax function: (used in multi-class classification)

$$\boxed{f(z_i) = \frac{e^{z_i}}{\Sigma_j e^{z_j}}}$$

→ converts outputs into probabilities (sum = 1)

Example for 3 classes (A, B, C):

i) $z_A = 2$, $z_B = 1$, $z_C = 0.5$

Softmax assigns: A = 57%, B = 29%, C = 14%

(c) Optimizers:

Optimizers adjust the neural network's weights to minimize the loss function.

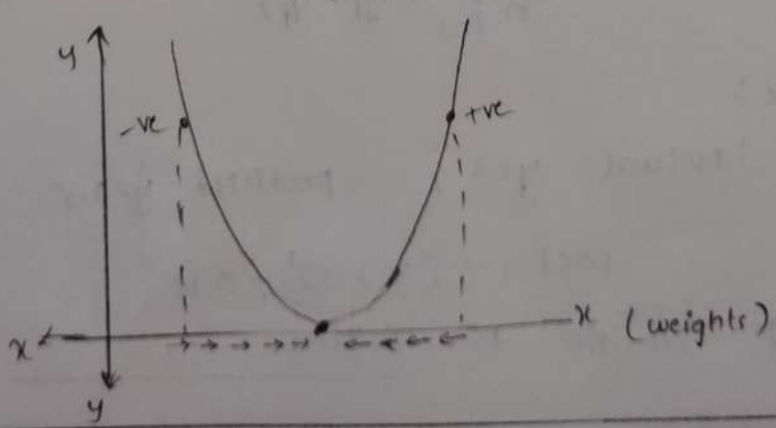
Types of optimizers:

1. Gradient Descent (Basic Method)

$$\boxed{w = w - \eta \frac{\partial Loss}{\partial w}}$$

$\eta$ = learning rate

problem: slow convergence.



2. Stochastic Gradient Descent (SGD)

- Updates weights using a random sample instead of the whole dataset.
- Faster than regular gradient descent but has high variance

3) Momentum - Based Gradient Descent:

- Uses momentum to accelerate learning and avoid getting stuck in local minima.

4) Adam (Adaptive moment estimation)

-> most widely used

$$w = w - \frac{\eta}{\sqrt{v_1} + \epsilon} m_t$$

-> combines momentum & adaptive learning rate

-> works well for most deep learning models.

(C) Loss functions: Loss function measure the error between actual and predicted values.

Types of loss functions:

1. Mean-Squared Error (MSE)

used for Regression

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Example:

- Actual $y = 3$, predicted $\hat{y} = 2.5$

$$MSE = (3-2.5)^2 = 0.25$$

problem: Sensitive to outliers.

POCO M6 PRO 5G 21/02/2025 21:03

2) Mean Absolute Error (MAE):

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

It is less sensitive to outliers than MSE

3) Cross-Entropy Loss (used in classification):

- for binary classification:

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

- for multi-class classification

$$Loss = -\sum_{i=1}^{n} y_i \log(\hat{y}_i)$$

Example: if actual class = 1, predicted probability = 0.9:

$$Loss = -\log(0.9) = 0.10$$

4) Huber Loss (Hybrid of MSE and MAE)

- used when outliers are present
- used MSE for small error and MAE for large errors.

POCO M6 PRO 5G 21/02/2025 21:03