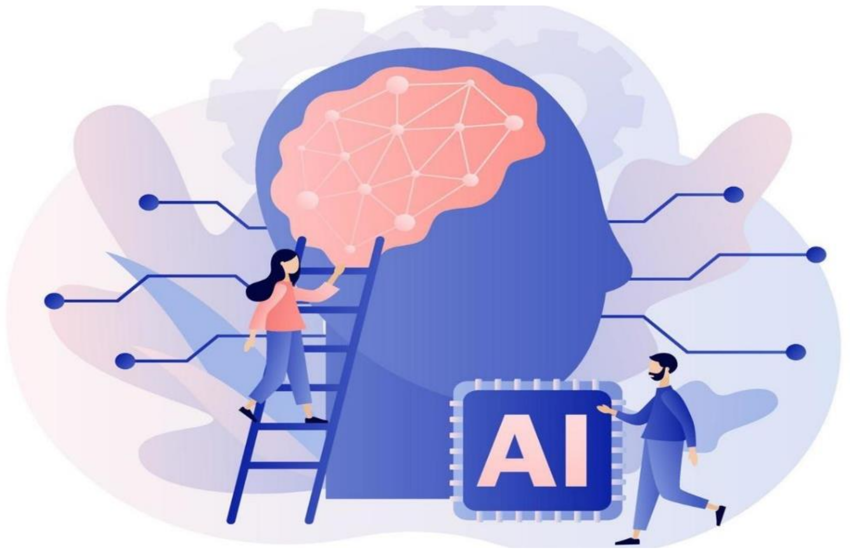
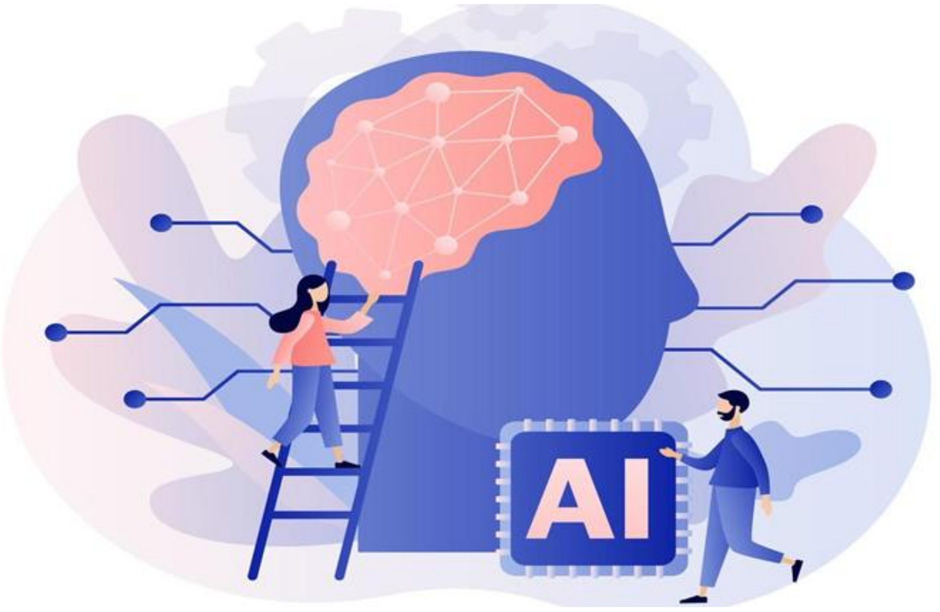


REVISION

Convolution Layer Calculations



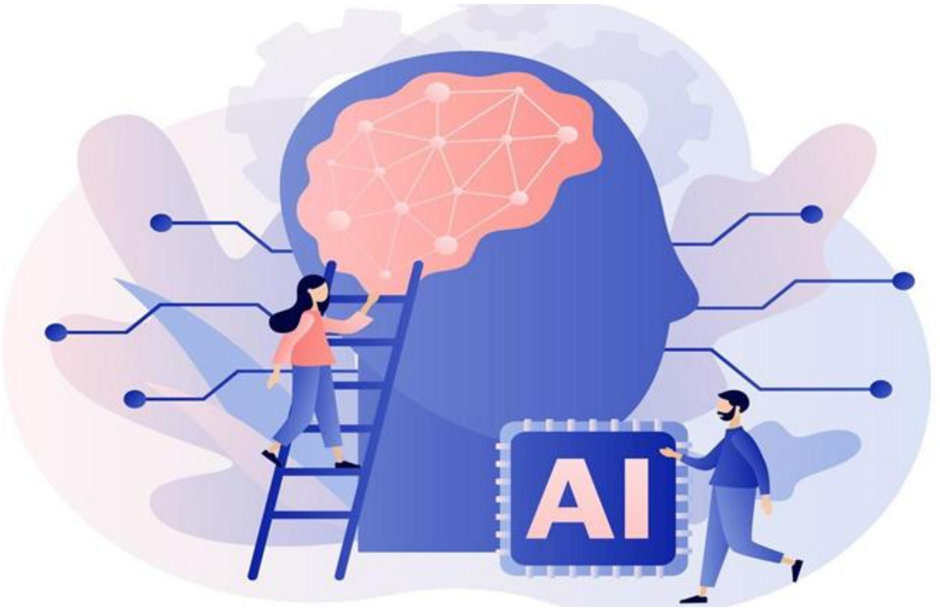
Example Q1:

An input image of size $64 \times 64 \times 3$ is passed through a convolutional layer with:

- 32 filters
- Filter size 3×3
- Stride = 1
- Padding = 1

👉 **Find the output shape.**

Convolution Layer Calculations



Example Q1 Sol:

An input image of size $64 \times 64 \times 3$ is passed through a convolutional layer with:

- 32 filters
- Filter size 3×3
- Stride = 1
- Padding = 1 ➔ padding = "same"

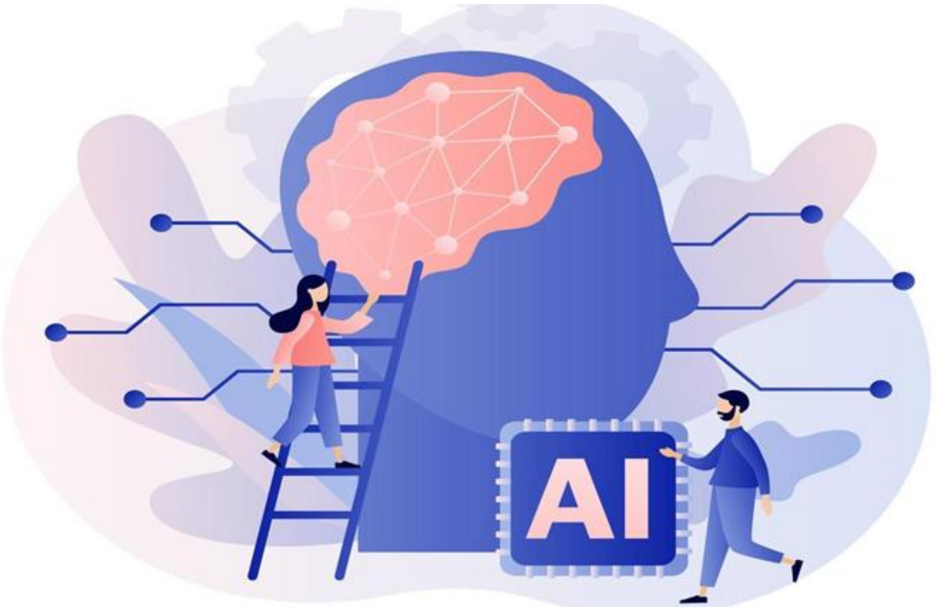
👉 **Find the output shape.**

$$O_w = \frac{(\mathbf{W} - \mathbf{F} + \mathbf{2P})}{S} + \mathbf{1} = \frac{\mathbf{64} - \mathbf{3} + \mathbf{2} * \mathbf{1}}{\mathbf{1}} + \mathbf{1} = \mathbf{64}$$

$$O_h = \frac{(\mathbf{H} - \mathbf{F} + \mathbf{2P})}{S} + \mathbf{1} = \frac{\mathbf{64} - \mathbf{3} + \mathbf{2} * \mathbf{1}}{\mathbf{1}} + \mathbf{1} = \mathbf{64}$$

O = $O_w * O_h * depth(\#of\ filters) =$ **64*64*32**

Convolution Layer Calculations



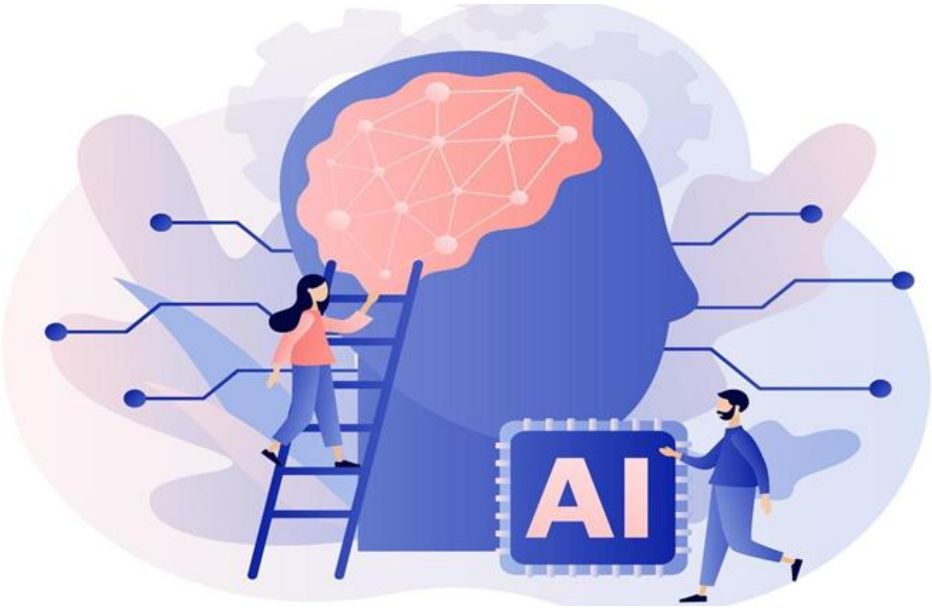
Example Q2:

An input image of size $64 \times 64 \times 3$ is passed through a convolutional layer with:

- 32 filters
- Filter size 3×3
- Stride = 2
- no padding

👉 **Find the output shape.**

Convolution Layer Calculations



Example Q2 Sol:

An input image of size $64 \times 64 \times 3$ is passed through a convolutional layer with:

- 32 filters
- Filter size 3×3
- Stride $= 2$
- no padding (padding $=$ "valid")

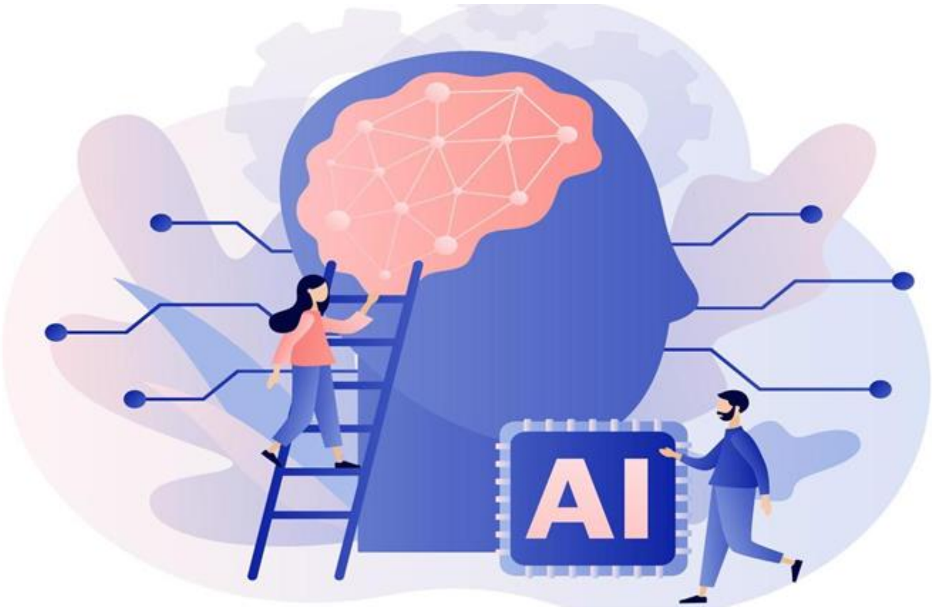
👉 **Find the output shape.**

$$O_w = \frac{(\mathrm{W} - \mathrm{F} + 2\mathrm{P})}{S} + 1 = \frac{64 - 3 + 2 * 0}{2} + 1 = 31.5 \cong 31$$

$$O_h = \frac{(\mathrm{H} - \mathrm{F} + 2\mathrm{P})}{S} + 1 = \frac{64 - 3 + 2 * 0}{2} + 1 = 31.5 \cong 31$$

O = $O_w * O_h * depth(\#of\ filters) =$ 31*31*32

Convolution Layer Calculations



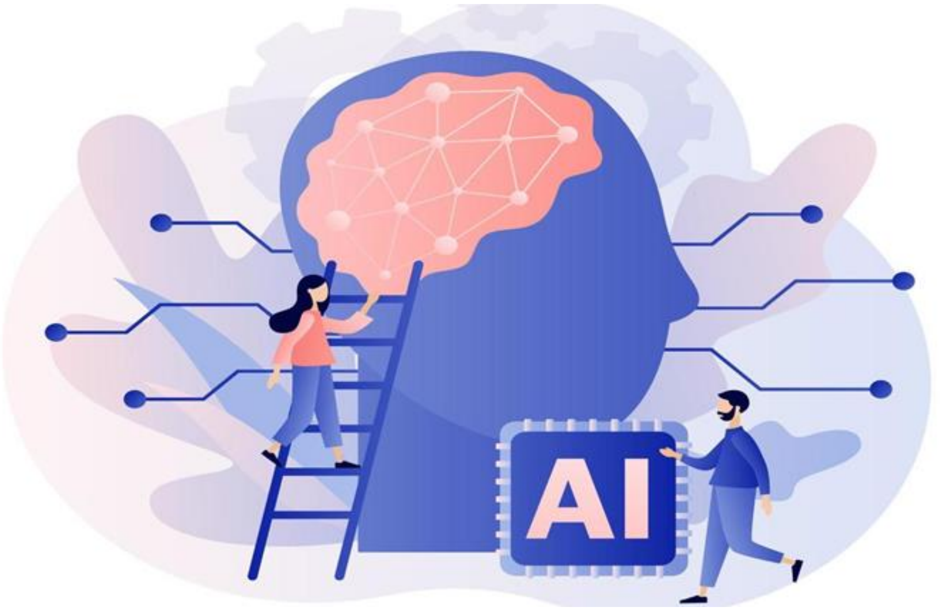
Example Q3:

An input image of size $32 * 32$ is passed through a convolutional layer with:

- 1 filters
- Filter size 5×5
- Stride = 1
- no padding

👉 **Find the output shape.**

Convolution Layer Calculations



Example Q3:

An input image of size $32 * 32$ is passed through a convolutional layer with:

- 1 filters
- Filter size 5×5
- Stride $= 1$
- no padding

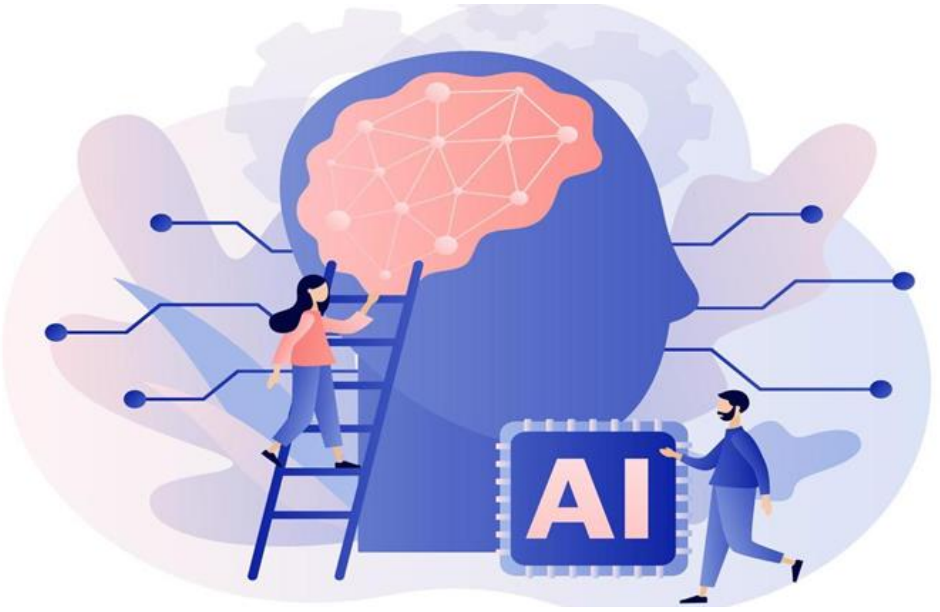
👉 **Find the output shape.**

$$O_w = \frac{(\mathrm{W}-\mathrm{F}+2\mathrm{P})}{S} + 1 = \frac{32-5+2*0}{1} + 1 = 28$$

$$O_h = \frac{(\mathrm{H}-\mathrm{F}+2\mathrm{P})}{S} + 1 = \frac{32-5+2*0}{2} + 1 = 28$$

$$\mathrm{O} = O_w * O_h * depth(\#of\ filters) = 28*28$$

Convolution Layer Calculations



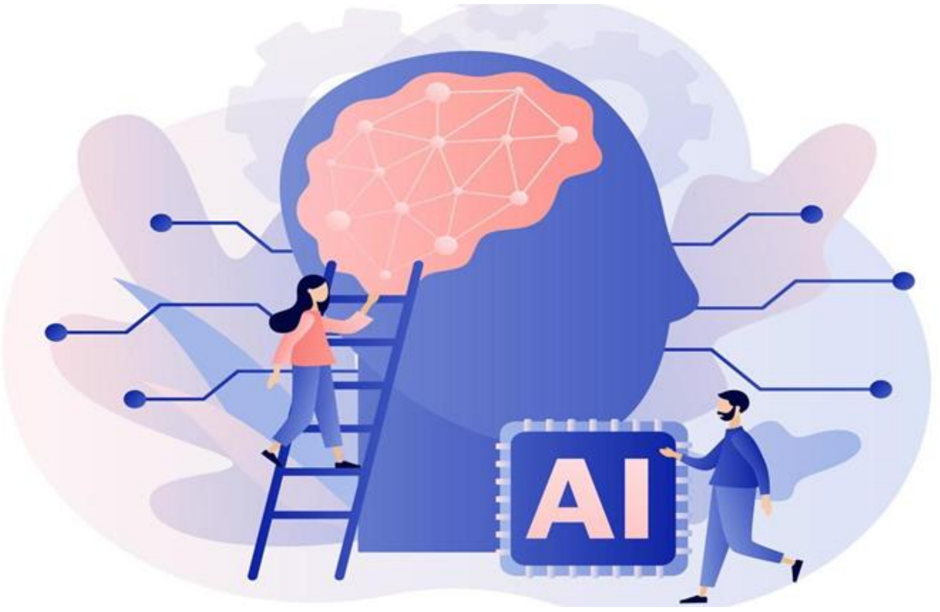
Example Q4:

An input image of size $28 * 28 * 64$ is passed through a convolutional layer with:

- 128 filters
- Filter size 3×3
- Stride = 1
- no padding

👉 **Find the The depth of a convolution kernel.**

Convolution Layer Calculations



Example Q4:

An input image of size $28 * 28 * 64$ is passed through a convolutional layer with:

- 128 filters
- Filter size 3×3
- Stride = 1
- no padding

Each convolution kernel (filter) must span all depth channels of the input feature map.

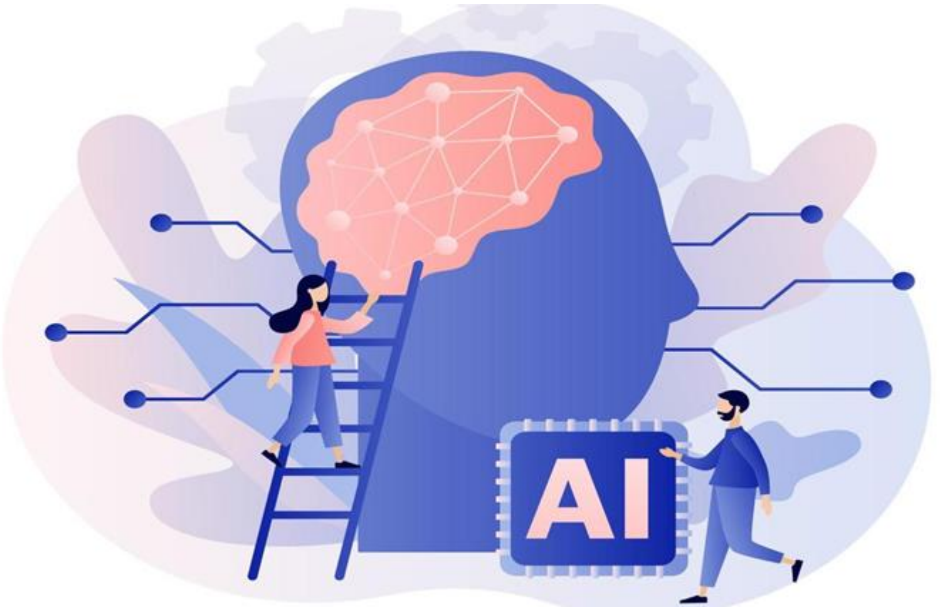
So even though the kernel size is 3 × 3, its depth must equal the input depth (64).

Depth = 64

👉 **Find the The depth of a convolution kernel.**

Depth of each convolution kernel: 64 / Number of kernels (output channels): 128

Convolution Layer Calculations



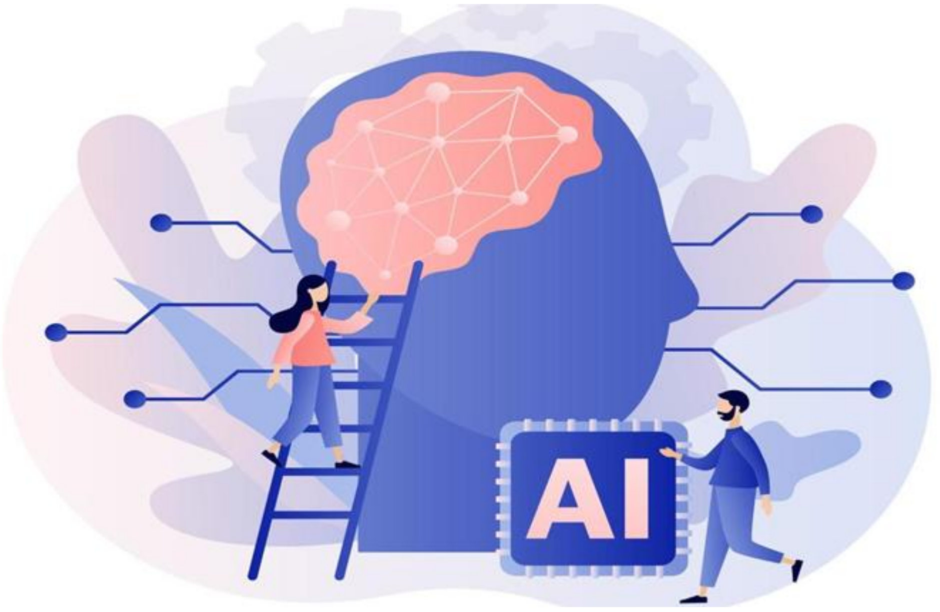
Example Q5:

An input image of size $64 * 64 * 16$ batch=8 is passed through a convolutional layer with:

- 64 filters
- Filter size 5×5
- Stride = 1
- Padding = “same”

👉 **Find the The output shape.**

Convolution Layer Calculations



Example Q5:

An input image of size $64 * 64 * 16$ batch=8 is passed through a convolutional layer with:

- 64 filters
- Filter size 5×5
- Stride = 1
- Padding = "same"

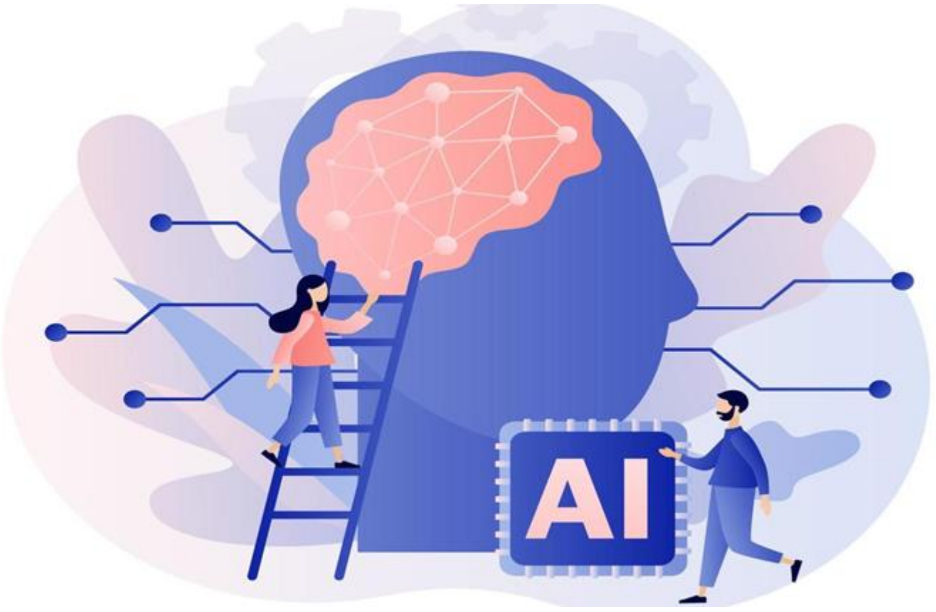
👉 **Find the The output shape.**

$$O_w = \frac{(W - F + 2P)}{S} + 1 = \frac{64 - 5 + 2 * 2}{1} + 1 = 64$$

$$O_h = \frac{(H - F + 2P)}{S} + 1 = \frac{64 - 5 + 2 * 2}{1} + 1 = 64$$

O = batch * $O_w * O_h * depth(\#of\ filters) =$ **8*64*64*64**

Number of Parameters in CNN Layer

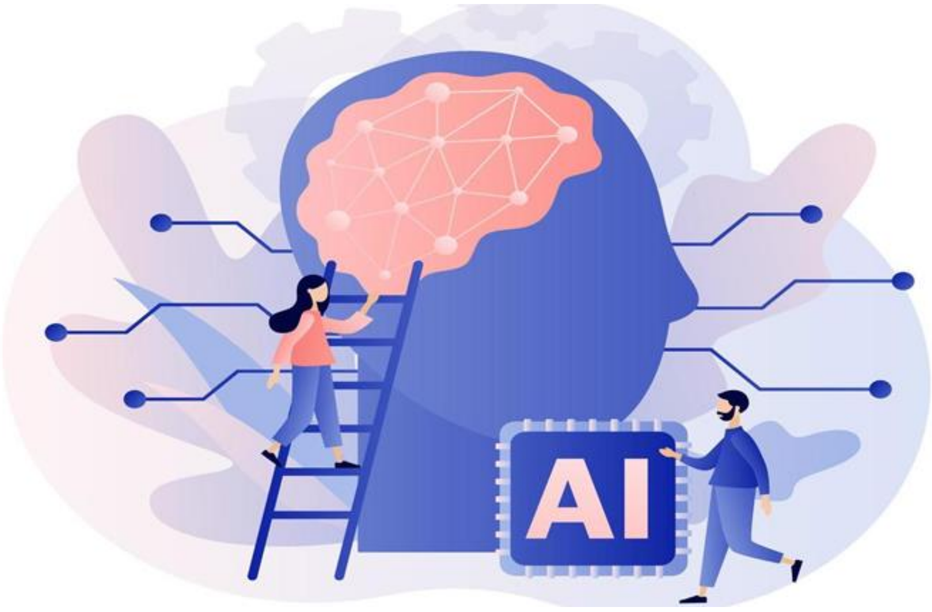


Example Q6:

Conv layer with 32 filters of size 3×3, input channels $= 3$.

👉 **Find number of parameters.**

Number of Parameters in CNN Layer



Example Q6 sol:

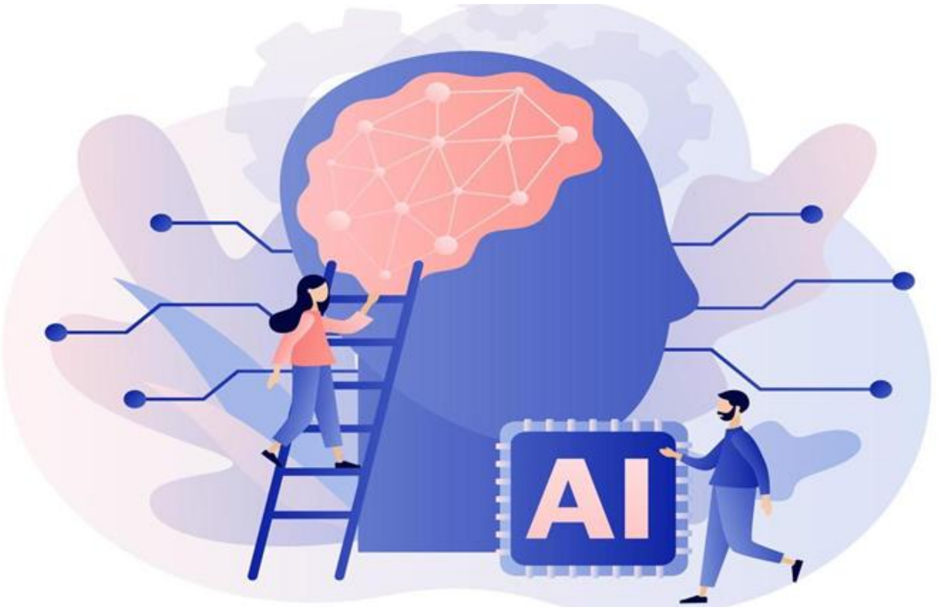
Conv layer with 32 filters of size 3×3, input channels $= 3$.

👉 **Find number of parameters.**

$$Params = (F_h * F_w * C_{in} + 1) * C_{out}$$

$$Params = (3 * 3 * 3 + 1) * 32 = 896$$

Number of Parameters in Dense Layer

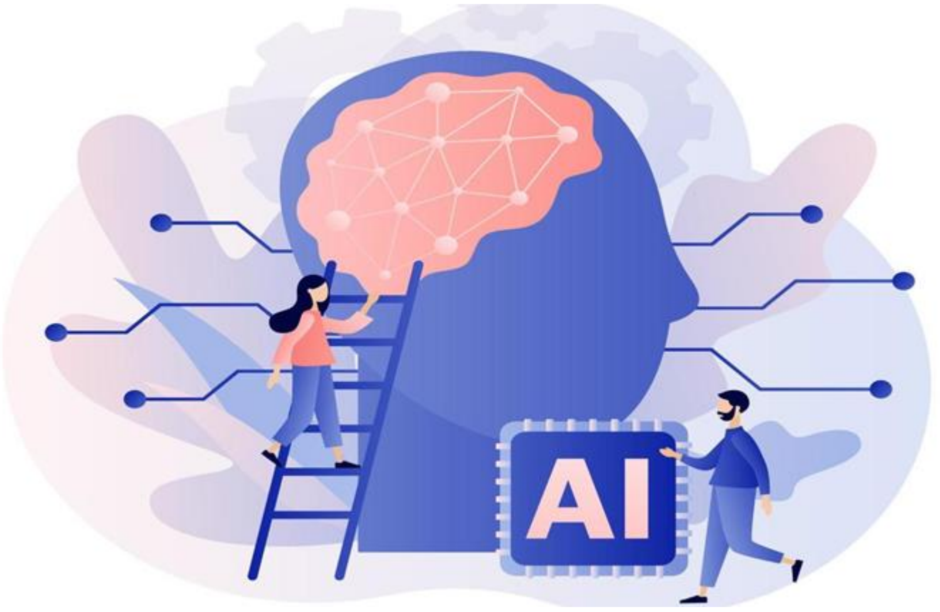


Example Q7:

Dense(128 → 10):

👉 **Find number of parameters.**

Number of Parameters in Dense Layer



Example Q7 sol:

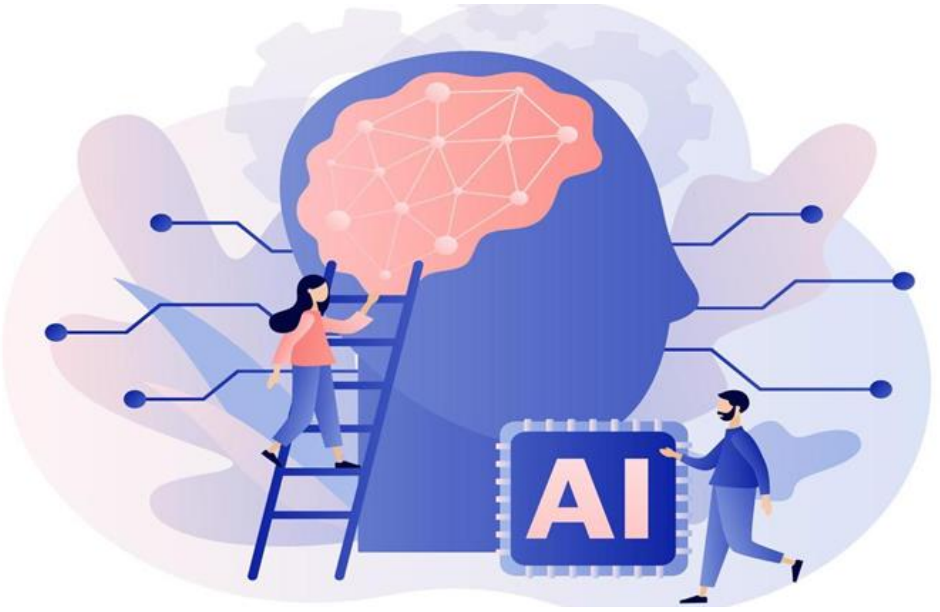
Dense(128 → 10):

👉 **Find number of parameters.**

$$\boldsymbol{Params = (\#of\ input\ neuron + bias) * \#of\ output\ neuron}$$

$$\boldsymbol{Params = (128 + 1) * 10 = 1290}$$

Loss Function Computations

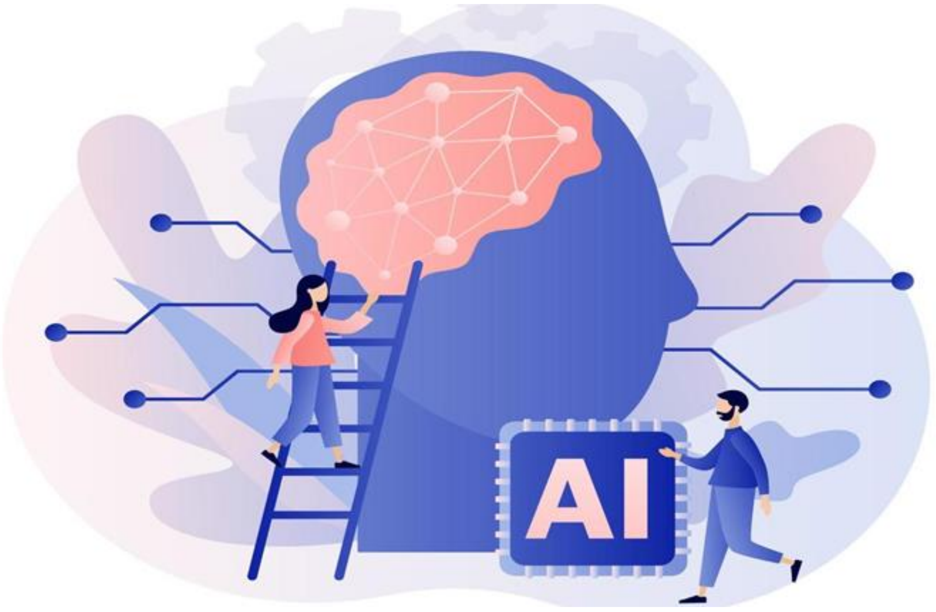


Example Q8:

$$y = [1, 0], \ \hat{y} = [0.8, 0.3]$$

👉 **Find MSE**

Loss Function Computations



Example Q8 sol:

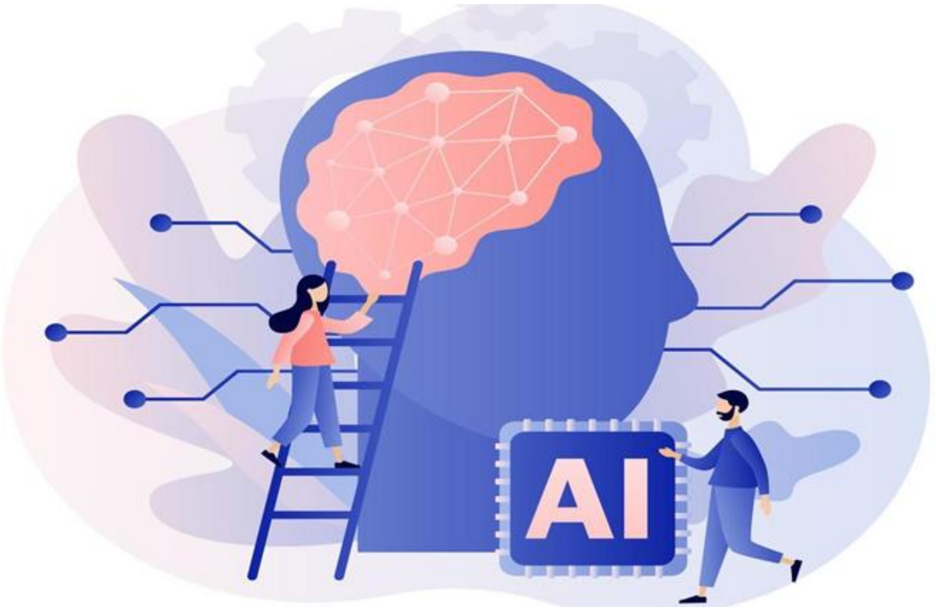
$$y = [1, 0],\ \hat{y} = [0.8, 0.3]$$

👉 **Find MSE**

$$MSE = \frac{1}{2}\sum (y - \hat{y})^2$$

$$MSE = \frac{(1 - 0.8)^2 + (0 - 0.3)^2}{2} = \frac{0.04 + 0.09}{2} = 0.065$$

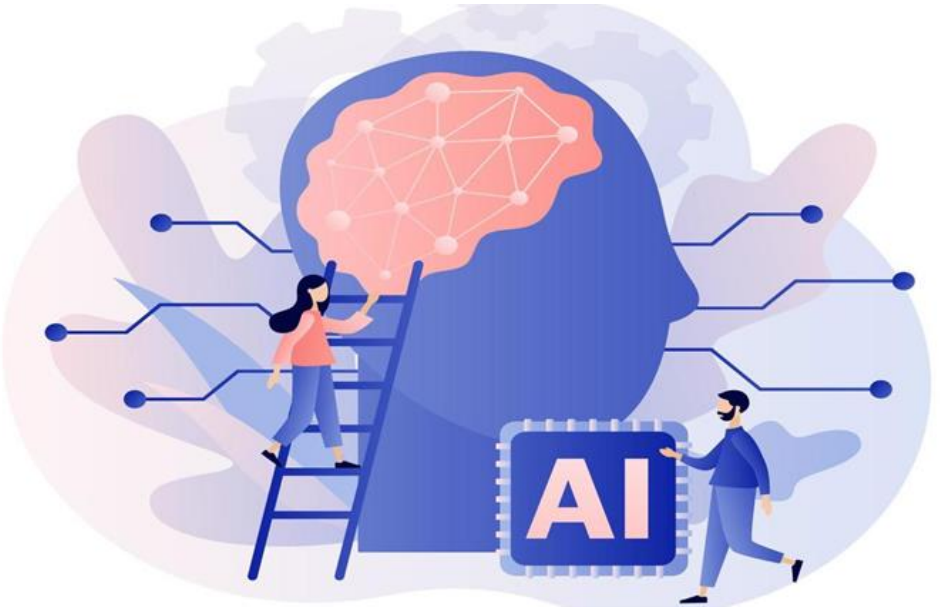
Loss Function Computations



Example Q9: Given three hypotheses (best_fit1, best_fit2, best_fit3) for linear regression, find the hypothesis with minimum cost function using MSE?

X	Y	Best_fit1	Best_fit2	Best_fit3
1	1	0.5	1	1.5
2	2.5	1	2	3
3	3.5	1.5	3	4

Loss Function Computations



Example Q9 sol: Given three hypotheses (best_fit1, best_fit2, best_fit3) for linear regression, find the hypothesis with minimum cost function using MSE?

X	Y	Best_fit1	Best_fit2	Best_fit3
1	1	0.5	1	1.5
2	2.5	1	2	3
3	3.5	1.5	3	4

$$MSE = \frac{1}{2}\sum(y-\hat{y})^2$$

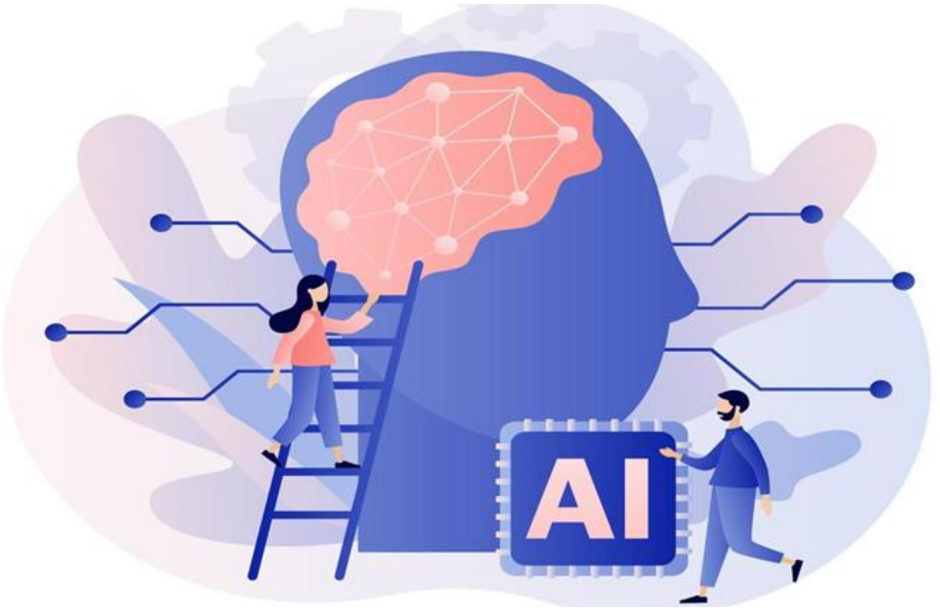
$$best_{fit1} = \frac{(1-0.5)^2+(2.5-1)^2+(3.5-1.5)^2}{3} = 2.167$$

$$best_{fit2} = \frac{(1-1)^2+(2.5-2)^2+(3.5-3)^2}{3} = 0.167$$

$$best_{fit3} = \frac{(1-1.5)^2+(2.5-3)^2+(3.5-4)^2}{3} = 0.25$$

So bestFit line is BestFit2

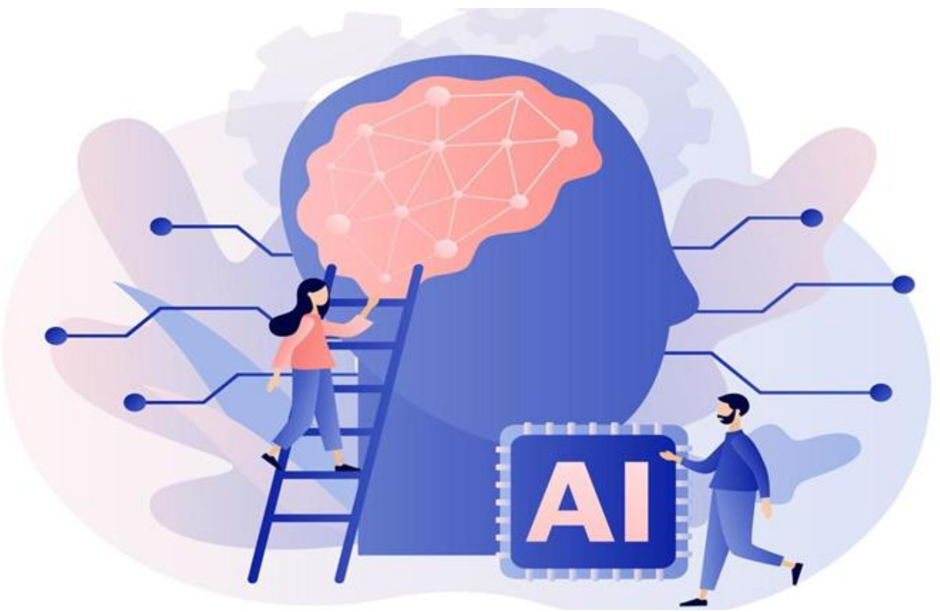
Loss Function Computations



Example Q9 another version: Given three hypotheses (best_fit1, best_fit2, best_fit3) for linear regression, find the hypothesis with minimum cost function using MSE and MAE?

X	Y	Best_fit1	Best_fit2	Best_fit3
1	2	1.8	2.0	2.2
2	1.5	1.2	1.7	1.4
3	3.0	2.9	3.2	3.1

Loss Function Computations



Example Q9 another version sol: Given three hypotheses (best_fit1, best_fit2, best_fit3) for linear regression, find the hypothesis with minimum cost function using MSE and MAE?

X	Y	Best_fit1	Best_fit2	Best_fit3
1	2	1.8	2.0	2.2
2	1.5	1.2	1.7	1.4
3	3.0	2.9	3.2	3.1

Best_fit1: predictions = [1.8, 1.2, 2.9]
Errors $y - \hat{y}$[0.1 ,0.3 ,0.2] =
Squared errors = [0.04, 0.09, 0.01] → sum = 0.14

$$\mathrm{MSE}_1 = \frac{0.14}{3} = 0.0467$$

Absolute errors = [0.2, 0.3, 0.1] → sum = 0.6

$$\mathrm{MAE}_1 = \frac{0.6}{3} = 0.2000$$

Best_fit2: predictions = [2.0, 1.7, 3.2]
Errors = [0.0, -0.2, -0.2]
Squared errors = [0.00, 0.04, 0.04] → sum = 0.08

$$\mathrm{MSE}_2 = \frac{0.08}{3} = 0.0267$$

Absolute errors = [0.0, 0.2, 0.2] → sum = 0.4

$$\mathrm{MAE}_2 = \frac{0.4}{3} \approx 0.1333$$

Best_fit3: predictions = [2.2, 1.4, 3.1]
Errors = [-0.2, 0.1, -0.1]
Squared errors = [0.04, 0.01, 0.01] → sum = 0.06

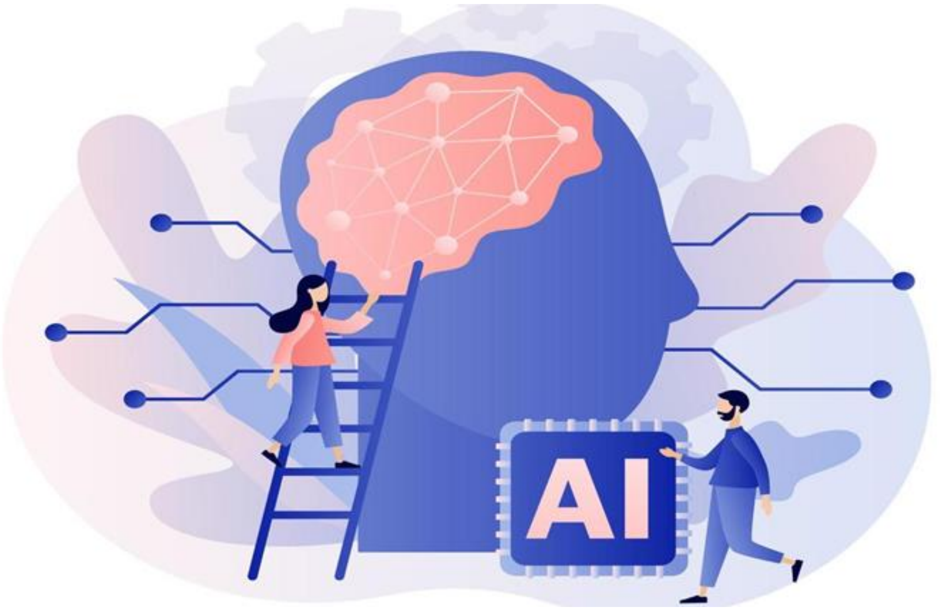
$$\mathrm{MSE}_3 = \frac{0.06}{3} = 0.0200$$

Absolute errors = [0.2, 0.1, 0.1] → sum = 0.4

$$\mathrm{MAE}_3 = \frac{0.4}{3} \approx 0.1333$$

Conclusions:
- **Best by MSE: Best_fit3** (MSE = 0.0200)
- **Best by MAE: Tie between Best_fit2 and Best_fit3** (MAE ≈ 0.1333)

Loss Function Computations

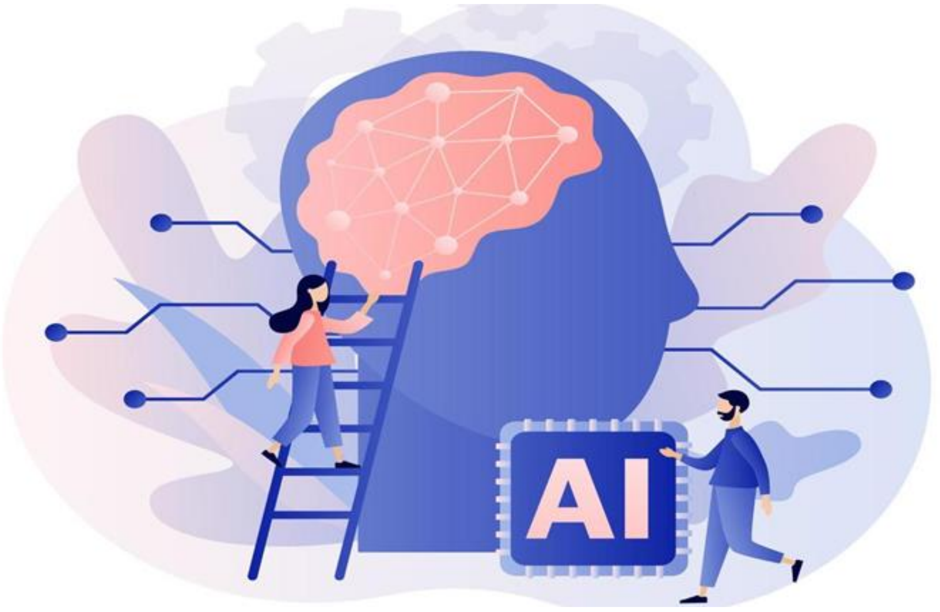


Example Q10:

$$y = [1, 0,0],\ \ \hat{y} = [0.7, 0.2,0.1]$$

Find Cross-Entropy Loss

Loss Function Computations



Example Q10:

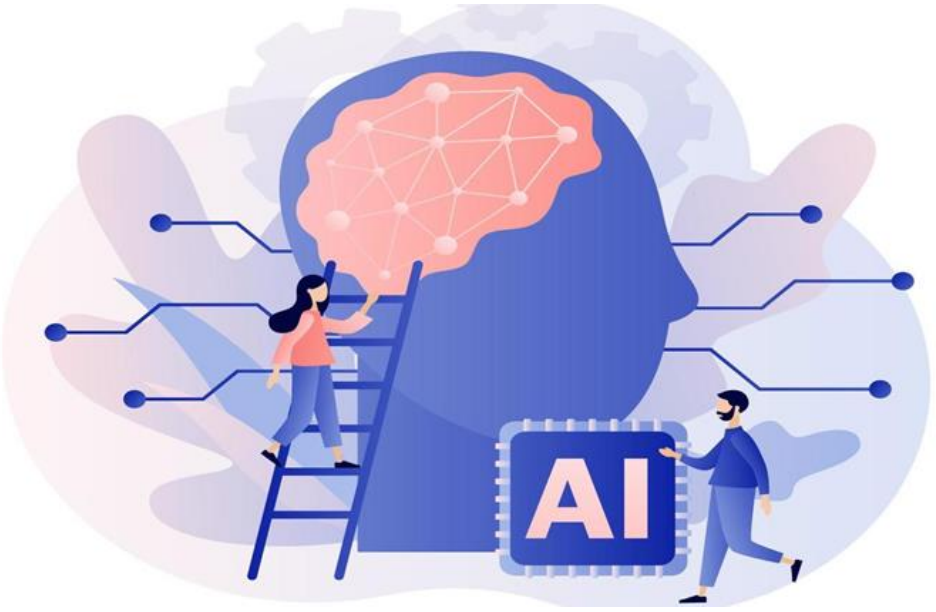
$$y = [1, 0,0],\ \hat{y} = [0.7, 0.2,0.1]$$

Find Cross-Entropy Loss

$$CrossEntropy = -\sum y_i log(\hat{y_i})$$

$$L = -log(0.7) = 0.357$$

Loss Function Computations

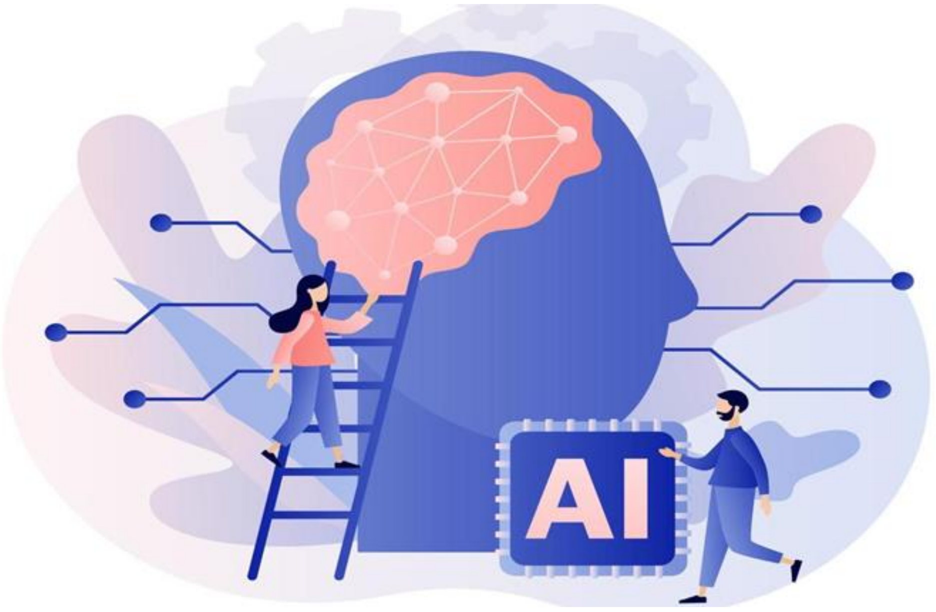


Example Q11:

True Label (y)	Predicted (ŷ)
1	0.9
0	0.2
1	0.4

Compute the **Binary Cross Entropy (BCE) loss.**

Loss Function Computations



Example Q11:

True Label (y)	Predicted (ŷ)
1	0.9
0	0.2
1	0.4

Sol:

1. For sample 1: y1=1,y^1=0.9 $L_1 = -[1 \cdot \log(0.9) + (0)\log(1-0.9)] = -\log(0.9) = 0.105$
2. For sample 2: y2=0,y^2=0.2 $L_2 = -[0 \cdot \log(0.2) + (1)\log(1-0.2)] = -\log(0.8) = 0.223$
3. For sample 3: y3=1,y^2=0.4 $L_3 = -[1 \cdot \log(0.4) + 0 \cdot \log(1-0.4)] = -\log(0.4) = 0.916$

- Now take the average: $\mathrm{BCE} = \frac{L_1 + L_2 + L_3}{3} = \frac{0.105 + 0.223 + 0.916}{3} = 0.4147$

Indicator variable

Prob of class *j*

$$-\sum_{j=1}^{M} y_j log(p(y_j))$$

Sum over trials

Sum over classes

$$-\sum_{i=1}^{N} y_i log(p(y_i)) + (1 - y_i) log(1 - p(y_i))$$

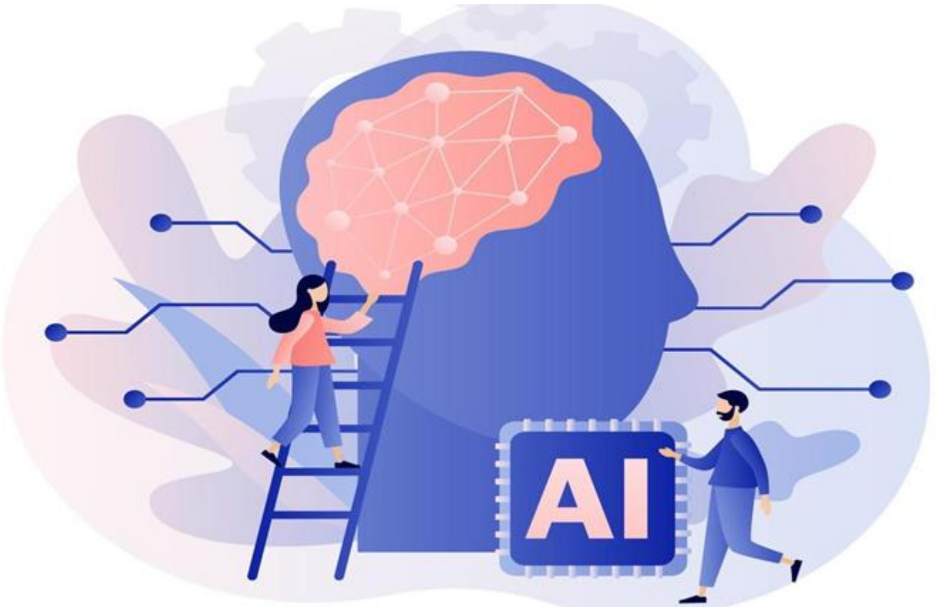
Label

Prob of positive class

Label

Prob of positive class

Loss Function Computations

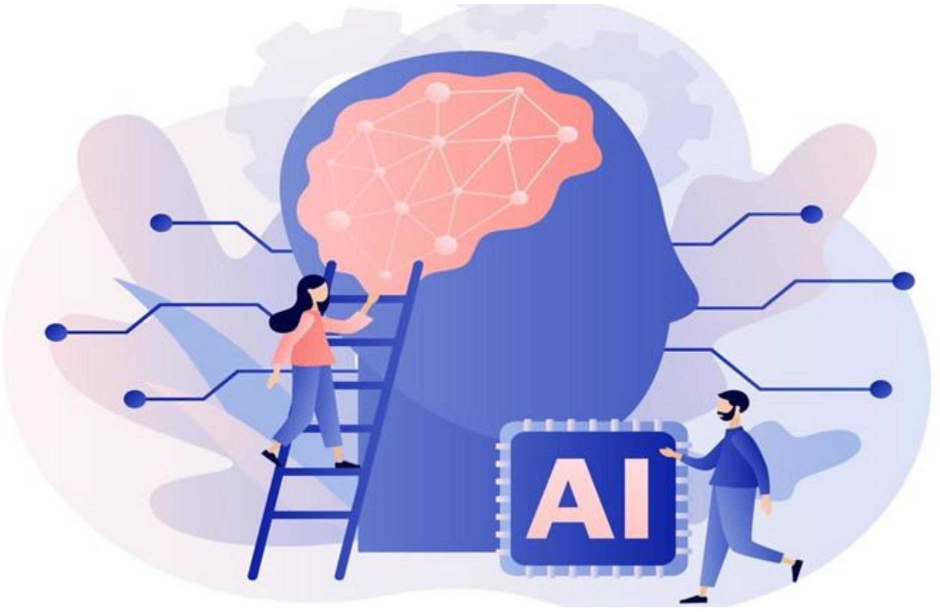


Example Q12:

If intersection = 80, $|A|=100$, $|B|=90$:

Find Dice Loss

Loss Function Computations



Example Q12:

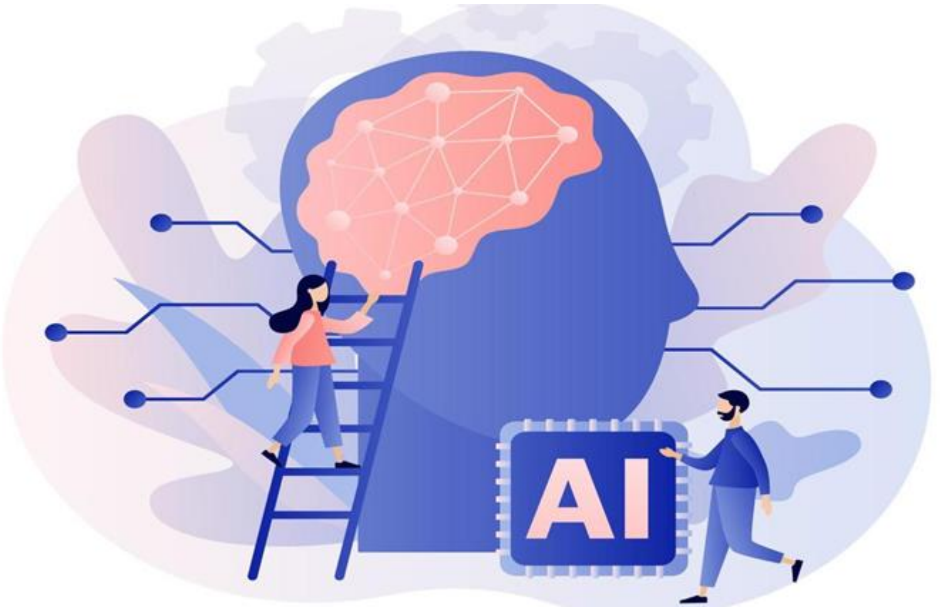
If intersection = 80, |A|=100, |B|=90:

👉 **Find** Dice Loss

$$\boldsymbol{Dice\ Coeff = \frac{2\,|A \cap B|}{|A| + |B|} = \frac{2 * 80}{100 + 90} = \frac{160}{190} = 0.842}$$

Loss = 1- $\boldsymbol{Dice\ Coeff = 1 - 0.842 = 0.158}$

Pooling Calculations

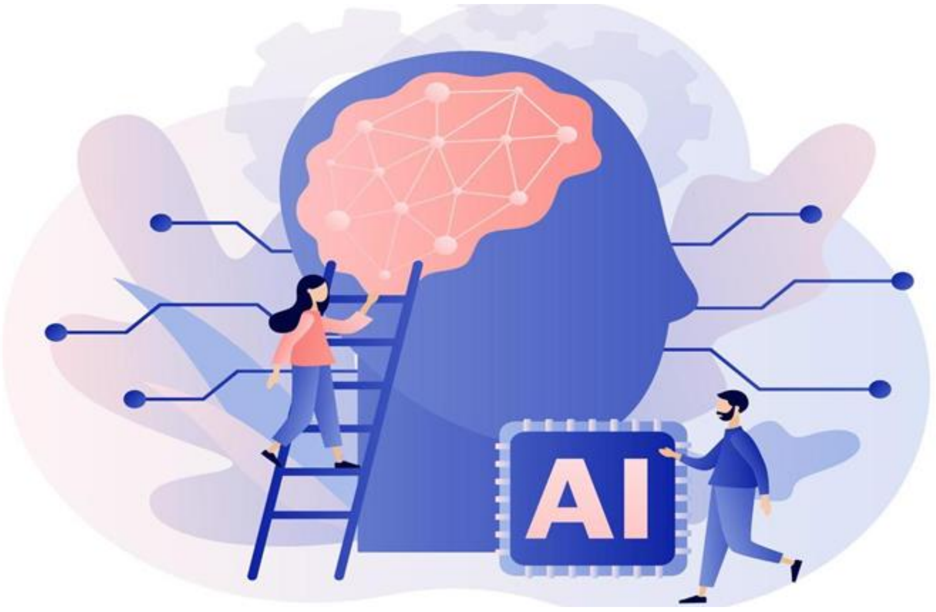


Example Q13:

Input: $32 \times 32 \times 64$, MaxPool 2×2, stride=2

👉 **Find** output shape

Pooling Calculations



Example Q13:

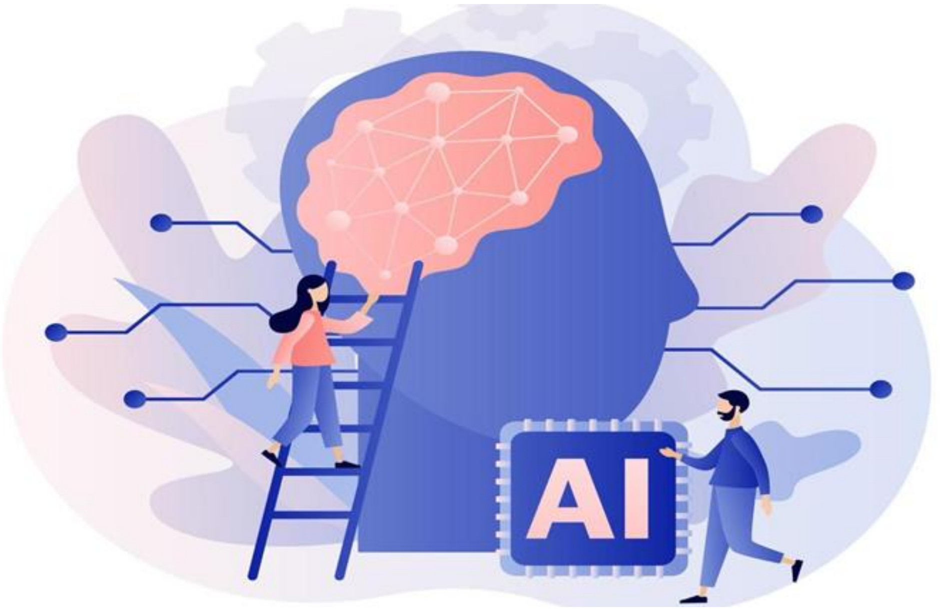
Input: $32 \times 32 \times 64$, MaxPool 2×2, stride=2

👉 **Find** output shape

$$\boldsymbol{O = \frac{32 - 2}{2} + 1 = 16}$$

Output shape = 16*16*64

Regularization Techniques

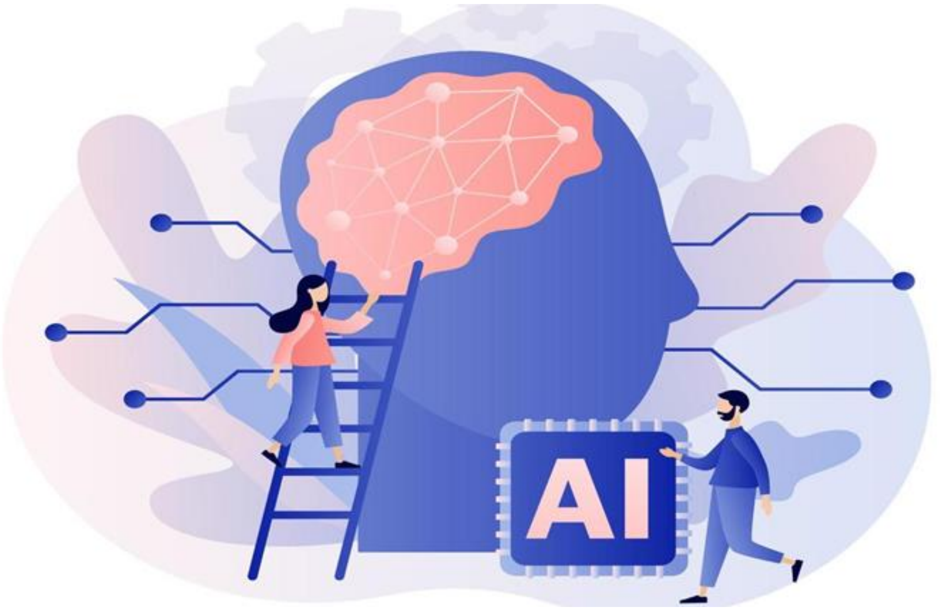


Example Q14:

Weights = [0.5, –0.2], λ = 0.01

👉 **Find** L1 Regularization

Regularization Techniques



Example Q14:

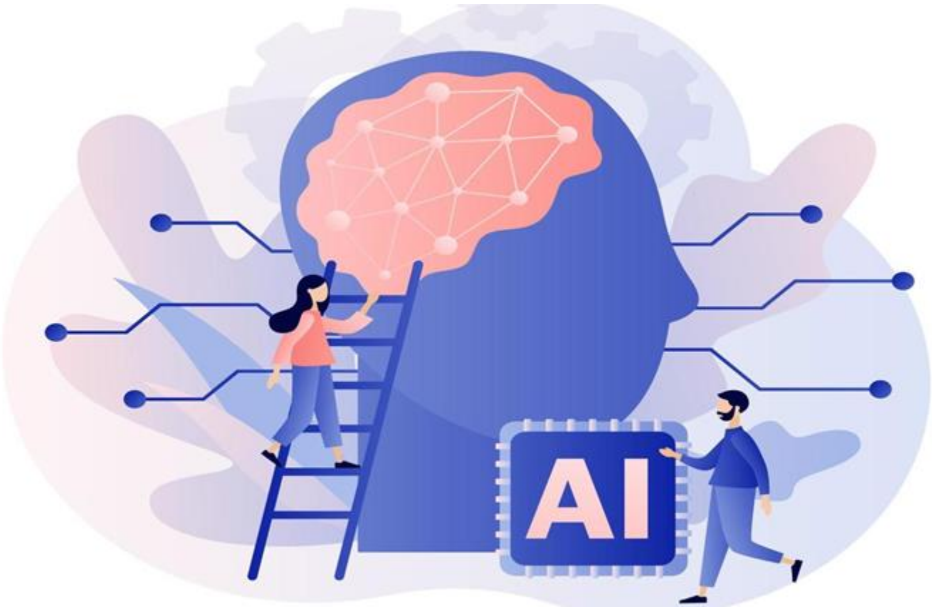
Weights = [0.5, –0.2], λ = 0.01

👉 **Find** L1,L2 term

$$L1 = L_{data} + \alpha \sum |w_i| = 0.01 * (0.5 + 0.2) = 0.007$$

$$L2 = L_{data} + \alpha \sum (w_i)^2 = 0.01 * \left((0.5)^2 + (-0.2)^2\right) = 0.01 * 0.29 = 0.0029$$

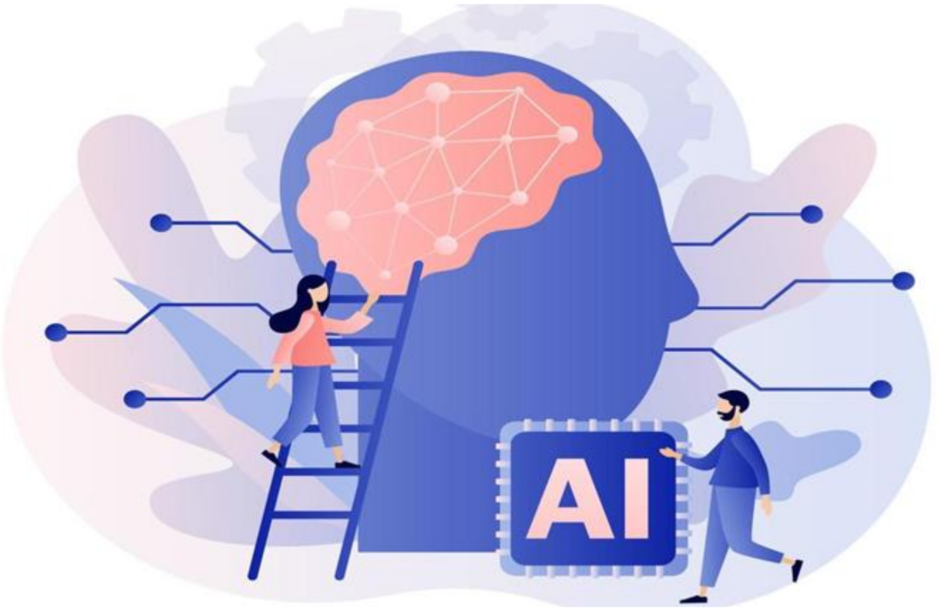
Vanishing / Exploding Gradients



Example Q15:

Why does using **ReLU** and **BatchNorm** help mitigate vanishing gradients?

Vanishing / Exploding Gradients

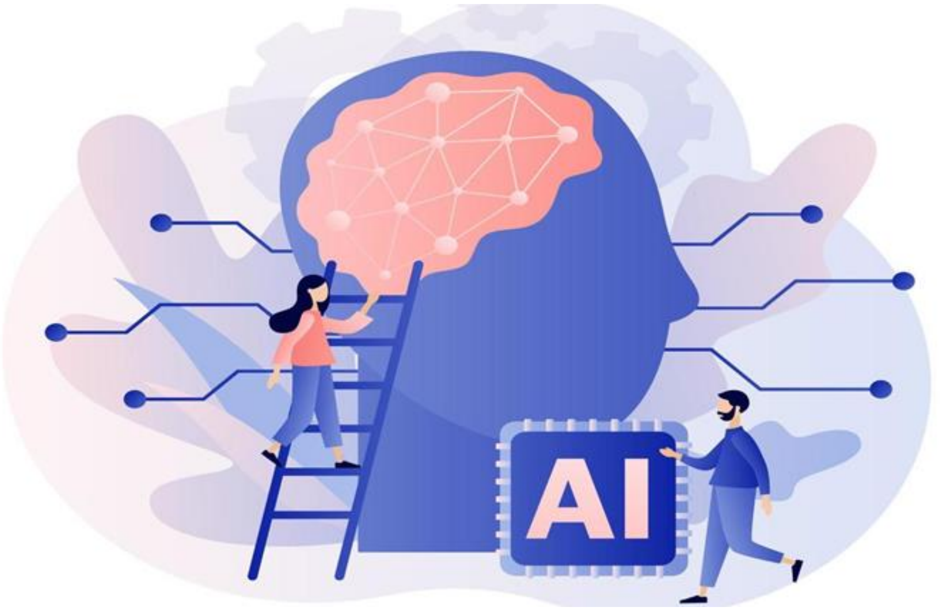


Example Q15:

Why does using **ReLU** and **BatchNorm** help mitigate vanishing gradients?

$\textit{Because ReLU keeps gradients active (non-saturating) and BN normalizes internal activations.}$

Loss Function Computations



Example Q16:

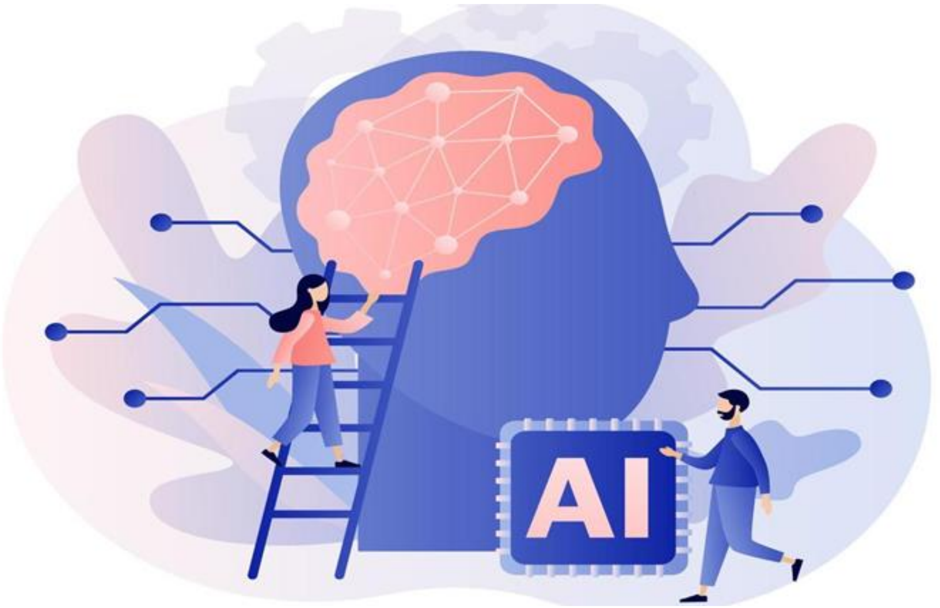
Sample	True Label (One-hot)	Predicted Probabilities y^
1	[1, 0, 0]	[0.7, 0.2, 0.1]
2	[0, 1, 0]	[0.1, 0.6, 0.3]
3	[0, 0, 1]	[0.2, 0.2, 0.6]

Find CrossEntropy Loss?

$$logloss = -\frac{1}{N}\sum_{i}^{N}\sum_{j}^{M} y_{ij}\log(p_{ij})$$

- N is the number of rows
- M is the number of classes

Loss Function Computations



Example Q16 sol:

Sample	True Label (One-hot)	Predicted Probabilities y^
1	[1, 0, 0]	[0.7, 0.2, 0.1]
2	[0, 1, 0]	[0.1, 0.6, 0.3]
3	[0, 0, 1]	[0.2, 0.2, 0.6]

$$logloss = -\frac{1}{N}\sum_{i}^{N}\sum_{j}^{M} y_{ij}\log(p_{ij})$$

- N is the number of rows
- M is the number of classes

Sol: N=3 and M=3

1. For sample 1: True label = class 1 → y=[1,0,0]
2. For sample 2: True label = class 2 → y=[0,1,0]
3. For sample 3: True label = class 3 → y=[0,0,1]
   - Now take the average:

$$L_1 = -[1 \cdot \log(0.7) + 0 \cdot \log(0.2) + 0 \cdot \log(0.1)] = -\log(0.7)$$
$$L_1 = 0.357$$

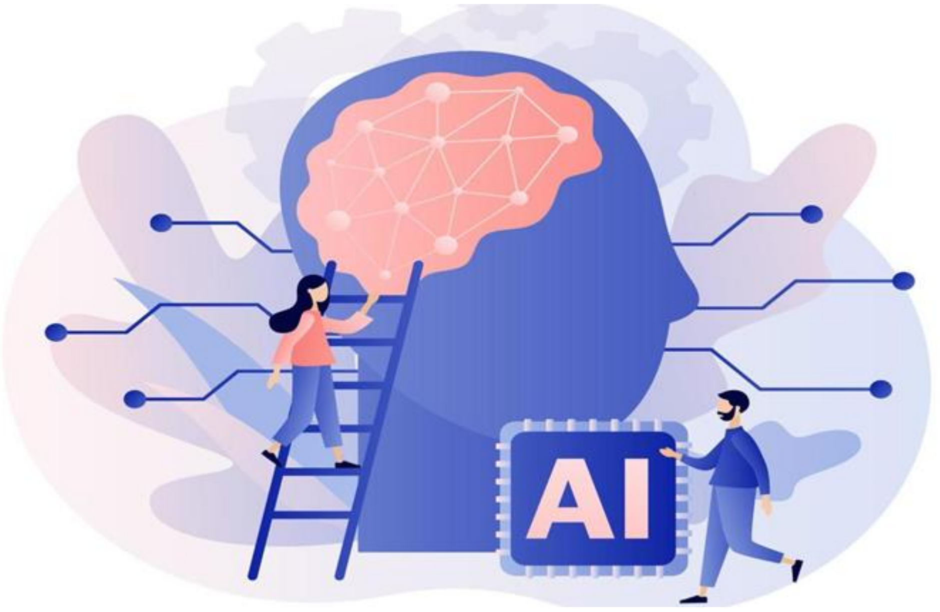
$$L_2 = -[0 \cdot \log(0.1) + 1 \cdot \log(0.6) + 0 \cdot \log(0.3)] = -\log(0.6)$$
$$L_2 = 0.511$$

$$L_3 = -[0 \cdot \log(0.2) + 0 \cdot \log(0.2) + 1 \cdot \log(0.6)] = -\log(0.6)$$
$$L_3 = 0.511$$

$$\text{CCE} = \frac{L_1 + L_2 + L_3}{3} = \frac{0.357 + 0.511 + 0.511}{3} = 0.4597$$

Loss Function Computations

Example Q17 : Question (Mathematical) Hinge Loss



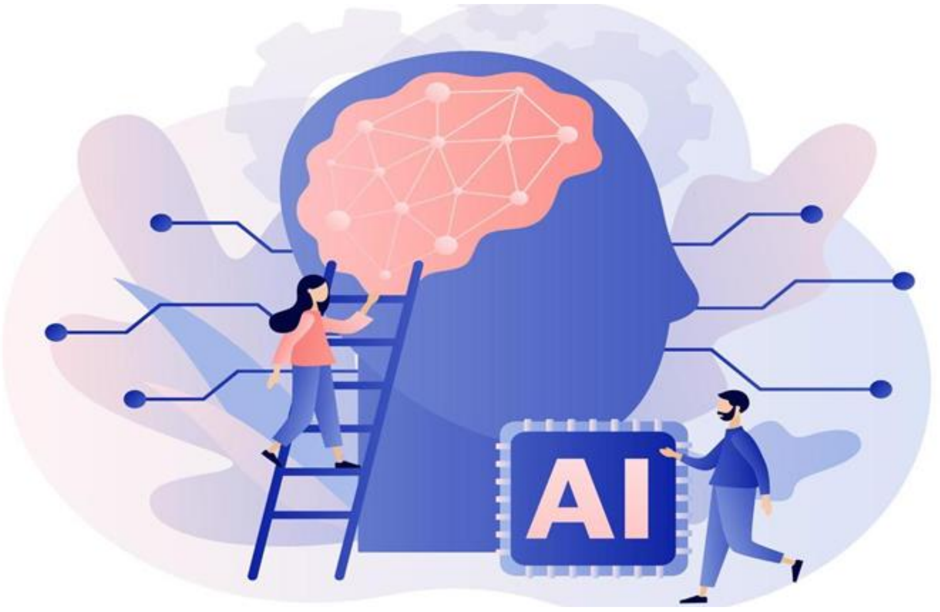
Example	True label y	Model output $\hat{y}$	$L = \max(0, 1 - y\hat{y})$	Interpretation
1	+1	+1.2	$\max(0, 1 - 1.2) = 0$	Correct & confident ✅
2	+1	+0.6	$\max(0, 1 - 0.6) = 0.4$	Correct but not confident enough ⚠
3	-1	+0.3	$\max(0, 1 - (-1)(0.3)) = \max(0, 1 + 0.3) = 1.3$	Wrong ❌

intended output $t = \pm 1$
class labels: +1 or -1

$$\ell(y) = \max(0, 1 - t \cdot y)$$

raw classification output

Mixed Challenge Questions



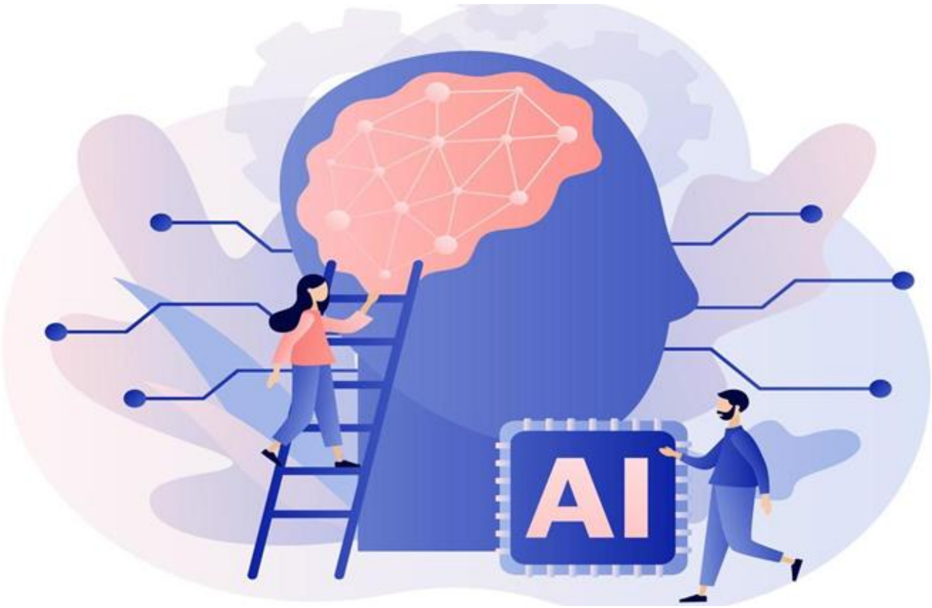
Example Q18:

An input of size $64 \times 64 \times 3$passes through:

- Conv: 3×3, stride=1, padding=1
- Then MaxPool: 2×2, stride=2
- Input: 64×64×3

👉 What is the output size after both layers?

Mixed Challenge Questions



Example Q18:

An input of size $64 \times 64 \times 3$passes through:

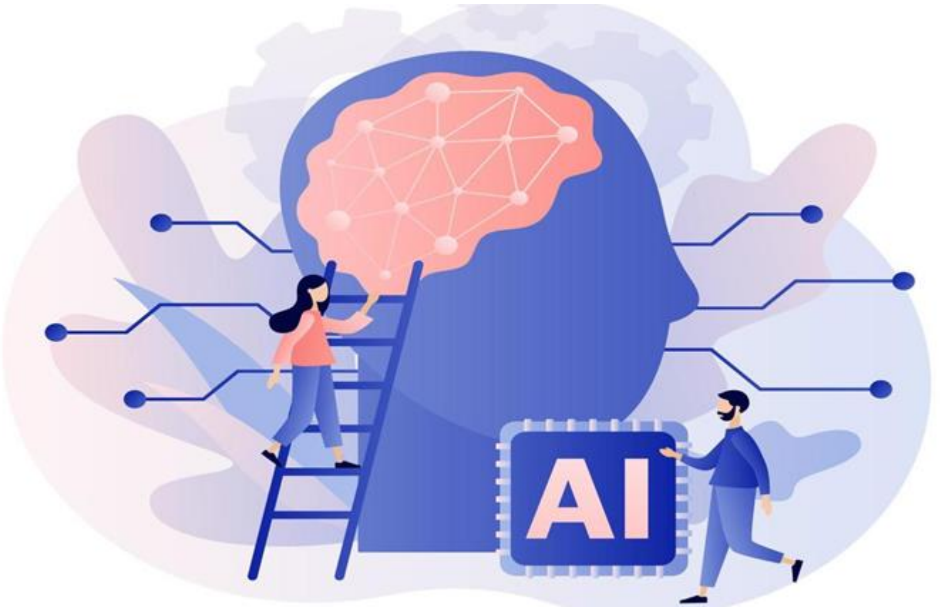
- Conv: 3×3, stride=1, padding=1
- Then MaxPool: 2×2, stride=2
- Input: 64×64×3

👉 What is the output size after both layers?

$$O1 = \frac{64 - 3 + 2}{1} + 1 = 64$$

O1 = 64*64*#filter

Mixed Challenge Questions



Example Q18:

An input of size $64 \times 64 \times 3$passes through:

- Conv: 3×3, stride=1, padding=1
- Then MaxPool: 2×2, stride=2
- Input: 64×64×3

👉 What is the output size after both layers?

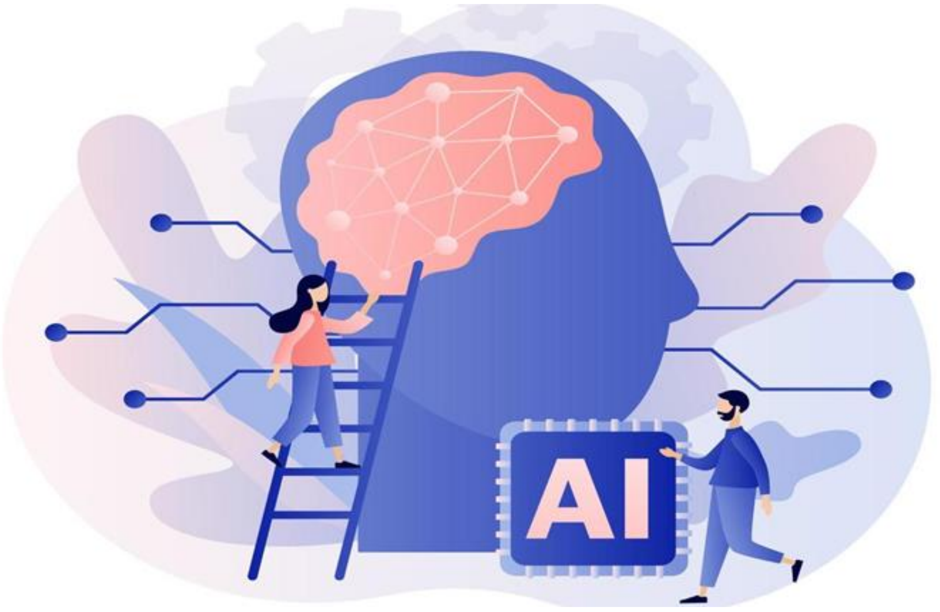
$$O1 = \frac{64 - 3 + 2}{1} + 1 = 64$$

O1 = 64*64*#filter

$$O2 = \frac{64 - 2}{2} + 1 = 32$$

O2 = 32*32*#filter

Mixed Challenge Questions

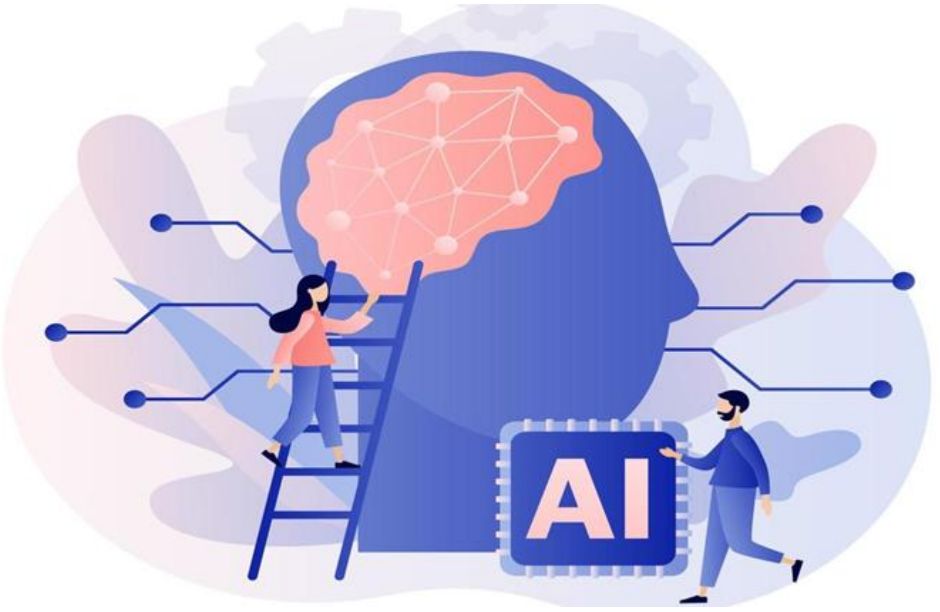


Example Q19:

Compute total trainable parameters in the following layers:

- Conv(3×3, 3→16)
- Conv(3×3, 16→32)
- Dense(512→10)

Mixed Challenge Questions

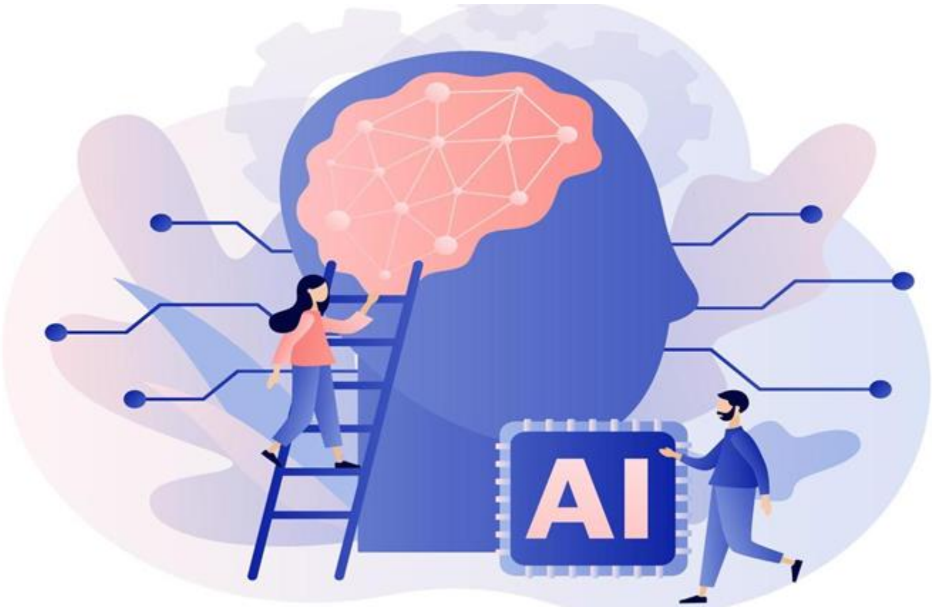


Example Q19:

Compute total trainable parameters in the following layers:

- Conv(3×3, 3→16)
- Conv(3×3, 16→32)
- Dense(512→10)

Conv1: $(3 \times 3 \times 3 + 1) \times 16 = 448$
Conv2: $(3 \times 3 \times 16 + 1) \times 32 = 4{,}640$
Dense: $(512 + 1) \times 10 = 5{,}130$
Total = 448 + 4,640 + 5,130 = 10,218 parameters



Thank You...!

End