Images are arrays of numbers

GIVEN

An 8-bit RGB image has shape 4 × 7 × 3.

Indexing is zero-based:
I[row, column, channel].

$$4 \times 7 \times 3$$

YOUR TASK

a. Count pixels and scalar values.

b. Is I[3, 7, 2] valid? Give the valid ranges.

c. Give the grayscale shape and value range.

d. Explain the channels and what grayscale loses.

Try the question before revealing the worked answer.

A1 · Count locations, then channels

PIXEL LOCATIONS

$$N_{\text{pixels}} = 4 \times 7 = 28$$

STORED SCALAR VALUES

$$N_{\text{values}} = 4 \times 7 \times 3 = 84$$

GRAYSCALE VERSION

$$4 \times 7 = 28 \text{ values}$$

Why 28 differs from 84

There are 28 spatial locations. Each stores three numbers, one per RGB channel. Counting channels again as pixels would confuse locations with stored values.

What grayscale changes

Grayscale stores one intensity per location, giving shape 4 × 7. An 8-bit unsigned value is an integer from 0 to 255.

A pixel is a location. An RGB pixel contains three scalar values.

A1 · Indexing and colour information

Axis	Length	Valid indices
Row	4	0, 1, 2, 3
Column	7	0, 1, 2, 3, 4, 5, 6
Channel	3	0, 1, 2

I[3, 7, 2] is invalid

Row 3 and channel 2 are allowed.
Column 7 lies beyond the final column.

Check each axis independently before accessing a pixel.

Counting starts at zero

An axis with n entries ends at n – 1. The length tells you how many entries exist, not the largest valid index.

Colour can disappear

RGB stores separate channel intensities. Grayscale combines them into brightness. Different colours can produce the same grayscale value, making them indistinguishable.

Does pixel similarity mean object similarity?

GIVEN

Use mean absolute difference (MAD).

A = [0, 0, 100, 0]
B = [0, 100, 0, 0]
C = [20, 20, 120, 20]

$$\mathrm{MAD}(A, B) = \frac{1}{N} \sum_{i=1}^{N} |A_i - B_i|$$

YOUR TASK

a. Calculate MAD(A, B) and MAD(A, C).

b. Which is numerically closer to A?

c. Describe the change in B and in C.

d. Why can viewpoint defeat pixel comparison?

Try the question before revealing the worked answer.

A2 · Work out each pixel difference

A COMPARED WITH B

$$|A - B| = [0, 100, 100, 0]$$

AVERAGE THE FOUR DIFFERENCES

$$\mathrm{MAD}(A, B) = \frac{200}{4} = 50$$

A COMPARED WITH C

$$\mathrm{MAD}(A, C) = \frac{20 + 20 + 20 + 20}{4} = 20$$

Why absolute values matter

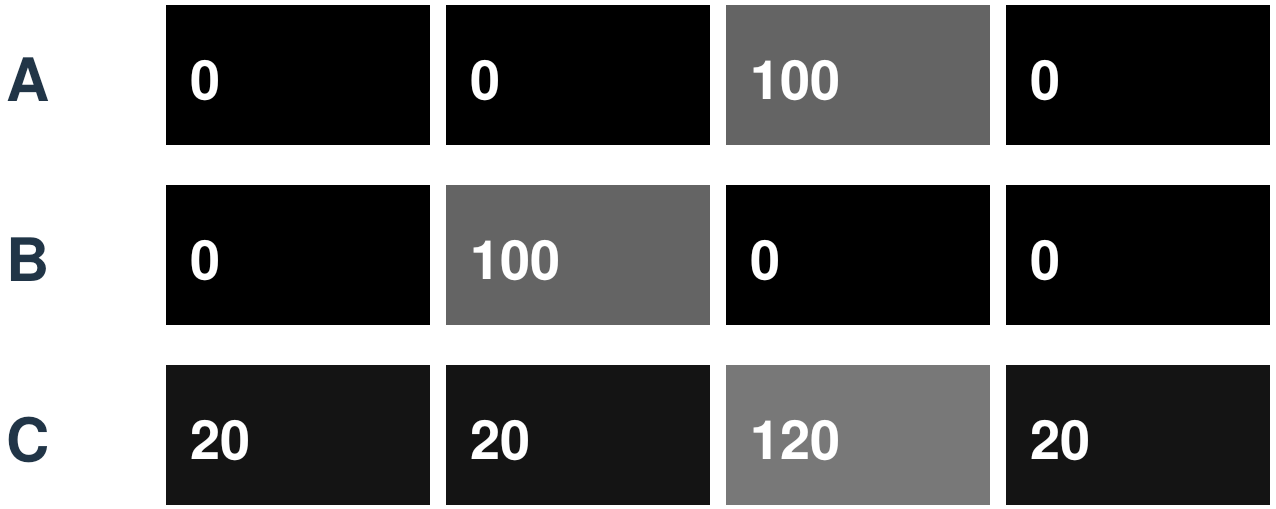
Subtract corresponding values, take each absolute value, then average. This prevents positive and negative differences from cancelling.

Interpret the ranking

C is closer because 20 < 50. The calculation compares aligned intensities. It does not recognize B as the same bright feature shifted to a new position.

Pixel distance measures intensity agreement at corresponding positions.

A2 · The same feature can move to different pixels



B shifts the bright feature one position left.

C adds 20 everywhere. Without clipping, this changes brightness but preserves pairwise intensity differences.

Why viewpoint is harder

Moving the camera changes projection, scale and visible surfaces. Occlusion can hide a region completely, so the same array index may describe a different object part.

Why local features can help

Descriptors summarize patterns near keypoints. Scale and rotation normalization can improve matching. Severe viewpoint changes and occlusion can still cause failure.

A change in pixels does not necessarily mean a change in object identity.

Brightness, contrast and clipping

GIVEN

Apply the transform to 8-bit intensities:
I = [40, 100, 180, 220].

For gamma, normalize to 0 ≤ u ≤ 1.

$$J = \text{clip}(1.5I + 10, 0, 255)$$

YOUR TASK

a. Why can lighting change a surface's RGB?

b. Transform all four values. Explain 1.5 and +10.

c. What information does clipping lose?

d. Using v = u^γ and u = 0.36, evaluate γ = 0.5 and γ = 2.

Try the question before revealing the worked answer.

A3 · Multiply, add, then clip

Input I	1.5I + 10	Clipped J
40	70	70
100	160	160
180	280	255
220	340	255

Compute first, clip second.

The multiplier controls intensity spread. The offset shifts the level.

Multiplication increases differences

Before clipping, every difference grows by 1.5. The difference 100 – 40 = 60 becomes 160 – 70 = 90. This increases contrast.

Addition shifts brightness

Adding 10 raises every value equally. Lighting, surface reflectance and camera settings also change recorded RGB, even when the physical surface stays the same.

A3 · Clipping explains why bright detail disappears

AN EQUIVALENT DEFINITION

$$\text{clip}(v, 0, 255) = \min(255, \max(0, v))$$

BELOW, INSIDE AND ABOVE THE RANGE

$$-20 \mapsto 0, \quad 160 \mapsto 160, \quad 280 \mapsto 255$$

TWO DIFFERENT INPUTS BECOME IDENTICAL

$$180 \mapsto 255, \qquad 220 \mapsto 255$$

Clipping is many-to-one: several inputs can share one output.

Why clip at 255?

An 8-bit unsigned value can store only 0 to 255. Clipping replaces an out-of-range value with its nearest valid endpoint.

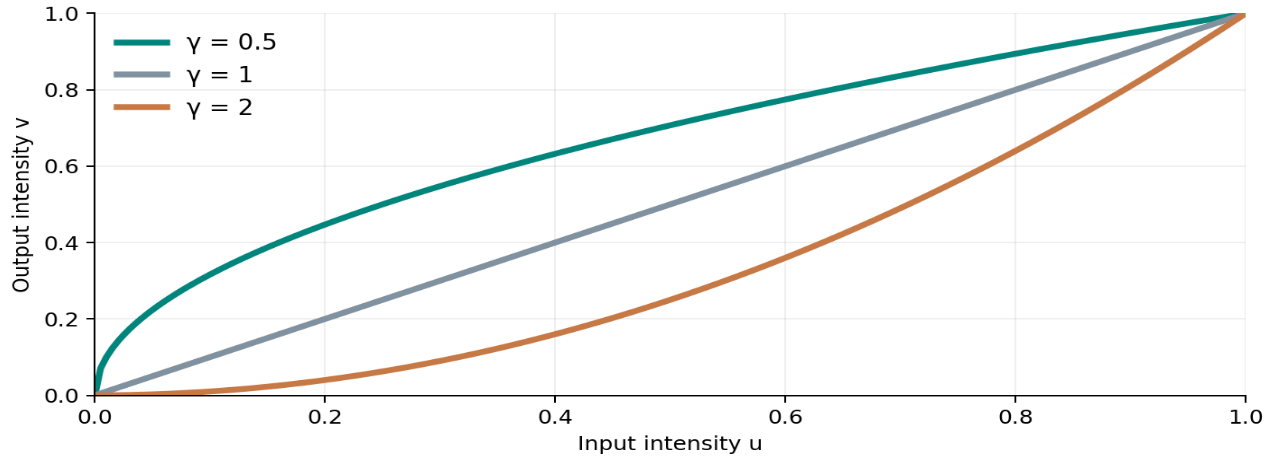
Why the loss is irreversible

Output 255 could come from 280, 340 or another high value. The clipped image cannot reveal the original value, so highlight texture may disappear.

A3 · Gamma changes intermediate values nonlinearly

$$u = \frac{I}{255}, \qquad v = u^{\gamma}$$

$$0.36^{0.5} = 0.6, \qquad 0.36^{2} = 0.1296$$



Why normalize?

The power-law rule is easiest to interpret on [0, 1]. Multiply the output by 255 to return to 8-bit units.

What gamma is used for

With this convention, γ < 1 brightens intermediate values and can reveal shadow detail. γ > 1 darkens them. Both keep 0 and 1 fixed, but cannot recover clipped detail.

Gamma changes brightness by different amounts at different input levels.

What can a histogram tell us?

GIVEN

Levels 0, 1, 2, 3 have counts:
[2, 1, 3, 2].

For equalization, use this exact rule:

$$J = \text{round}(7\,\text{CDF}(I))$$

YOUR TASK

a. Find the normalized histogram and CDF.

b. Where does input level 1 map?

c. What survives a shuffle? Can boundaries be located?

d. Stretch [40, 120] to [0, 255]. Where does 80 map?

Try the question before revealing the worked answer.

A4 · Normalize counts, then accumulate them

Level	Count	Probability	CDF
0	2	2/8 = 0.250	0.250
1	1	1/8 = 0.125	0.375
2	3	3/8 = 0.375	0.750
3	2	2/8 = 0.250	1.000

$$J(1) = \text{round}(7 \times 0.375) = 3$$

The CDF gives the fraction of pixels at or below a given intensity.

Normalize by the total

There are 8 pixels. Dividing each count by 8 gives probabilities that sum to 1.

The CDF includes lower levels

At level 1, add the probabilities of levels 0 and 1: 0.250 + 0.125 = 0.375. The given rule maps it to 3. Other equalization conventions can differ.

A4 · Contrast stretching preserves order

NORMALIZE THE INPUT RANGE

$$t = \frac{I - I_{\min}}{I_{\max} - I_{\min}}$$

SCALE TO THE OUTPUT RANGE

$$J = 255\,\frac{I - 40}{120 - 40}$$

SUBSTITUTE THE MIDPOINT

$$J(80) = 255\,\frac{40}{80} = 127.5 \approx 128$$

What a shuffle preserves

A histogram counts how often values occur. Reordering pixels leaves those counts unchanged. It cannot locate boundaries because it does not retain positions.

What stretching achieves

80 is halfway from 40 to 120, so it maps halfway from 0 to 255. This makes existing differences easier to see but cannot reconstruct missing information.

Histograms describe intensity distribution, not object arrangement.

Convolution and correlation: which kernel order?

GIVEN

Window: [3, 5, 8]
Kernel: [2, −1, 0]

For image size, use a 3 × 3 kernel with stride 1.

$$H_{\text{out}} = H + 2p - k + 1$$

YOUR TASK

a. Calculate cross-correlation.

b. Calculate convolution and explain the difference.

c. When do the two operations agree?

d. For a 6 × 6 image, give output sizes for p = 0 and p = 1.

Try the question before revealing the worked answer.

A5 · The flip changes which values are multiplied

CROSS-CORRELATION: KEEP THE KERNEL ORDER

$$3(2)+5(-1)+8(0)=6-5=1$$

CONVOLUTION: REVERSE THE KERNEL

$$[2,-1,0] \longmapsto [0,-1,2]$$

MULTIPLY AND ADD AGAIN

$$3(0)+5(-1)+8(2)=0-5+16=11$$

Always settle the kernel orientation before calculating the response.

Why the outputs differ

The kernel is asymmetric. Reversing it changes the coefficients applied to the first and last pixels. The centre coefficient stays –1.

How this extends to images

In two dimensions, convolution flips the kernel horizontally and vertically. Cross-correlation applies the kernel as written. Check which convention a question specifies.

A5 · Symmetry and padding determine the remaining cases

NO PADDING

$$H_{\text{out}} = 6 + 2(0) - 3 + 1 = 4$$

ONE PIXEL OF PADDING PER SIDE

$$H_{\text{out}} = 6 + 2(1) - 3 + 1 = 6$$

APPLY THE SAME RULE TO WIDTH

$$4 \times 4 \quad \text{or} \quad 6 \times 6$$

When the operations agree

If flipping leaves the kernel unchanged, convolution and correlation produce the same result under matching boundary rules. A symmetric Gaussian is an example.

Why padding matters

Without padding, a 3 × 3 window fits only at four positions along each side. Padding adds border values and permits six positions. Those extra values also affect border responses.

Same output size does not mean unchanged border behaviour.

Smoothing: noise or a genuine small feature?

GIVEN

A 3 × 3 grayscale patch has centre 82.
All eight neighbours equal 10.

Mean filter: every coefficient is 1/9.
Gaussian: [1 2 1; 2 4 2; 1 2 1]/16.

YOUR TASK

a. Find the mean-filter centre response.

b. Find the Gaussian centre response.

c. Why preserve a constant patch? Why weight nearby pixels more?

d. Explain the result if 82 is noise, then if it is a real feature.

Try the question before revealing the worked answer.

A6 · Compare the two weighted averages

10	10	10
10	82	10
10	10	10

Input patch

1/16	2/16	1/16
2/16	4/16	2/16
1/16	2/16	1/16

Gaussian weights

$$J_{\text{mean}} = \frac{82 + 8(10)}{9} = 18$$

$$J_{\text{Gaussian}} = \frac{4(82) + 12(10)}{16} = 28$$

The mean treats every pixel equally

Nine values sum to 162. Dividing by 9 replaces the bright centre with 18, close to its dark neighbours.

The Gaussian favours the centre

The centre has weight 4/16. All remaining weights sum to 12/16. Its response is therefore 28, retaining more of the central intensity.

A filter response is a weighted sum, not a decision about whether a feature is real.

A6 · Smoothing trades noise reduction for detail

A CONSTANT PATCH HAS VALUE c EVERYWHERE

$$J = \sum_{u,v} K_{u,v}\, c$$

A NORMALIZED KERNEL HAS UNIT SUM

$$\sum_{u,v} K_{u,v} = 1$$

THEREFORE THE RESPONSE STAYS CONSTANT

$$J = c \sum_{u,v} K_{u,v} = c$$

When smoothing helps

If 82 is a noise spike, mixing it with neighbours suppresses the spike. Gaussian weights favour nearby evidence. Increasing the blur scale usually removes more noise and detail.

When smoothing harms

If 82 represents a tiny real object, the same averaging weakens that object. Edges also blur. Constant-value preservation applies where the full patch is constant, including the boundary treatment.

A sum of one preserves constants; by itself it does not prove a kernel smooths.

Gradients describe rapid intensity changes

GIVEN

At one pixel:
$G_x = 6$ and $G_y = 8$.

Coordinates: x rightward, y downward.
Use atan2 to determine the direction.

$$\theta = \text{atan2}(G_y, G_x)$$

YOUR TASK

a. Calculate gradient magnitude and direction.

b. How is the edge oriented relative to the gradient?

c. Why does differentiation amplify noise?

d. Explain spatial/frequency views and low-/high-pass filtering.

Try the question before revealing the worked answer.

A7 · Gradient magnitude and direction

MAGNITUDE COMBINES BOTH DERIVATIVES

$$\|\nabla I\| = \sqrt{G_x^2 + G_y^2} = \sqrt{36 + 64} = 10$$

DIRECTION USES BOTH SIGNS

$$\theta = \operatorname{atan2}(8, 6) \approx 53.1^\circ$$

THE EDGE TANGENT IS PERPENDICULAR

$$\theta_{\text{edge}} \approx 53.1^\circ + 90^\circ = 143.1^\circ$$

The gradient points across an edge, not along it.

What the numbers mean

Magnitude 10 describes the local strength of intensity change. The gradient points toward the strongest increase, from +x toward +y using the stated coordinate convention.

Why the edge is perpendicular

An edge follows a line of roughly constant intensity. Moving across it changes intensity most strongly. Edge orientation is equivalent modulo 180°, so reversing the tangent gives the same edge.

A7 · Frequency explains the noise–detail trade-off

View or filter	What it emphasizes
Spatial image	Intensity at each position
Low frequencies	Slowly varying intensity
High frequencies	Rapid change, edges, fine detail
Low-pass filter	Smooth variation, less fine detail
High-pass filter	Edges and other rapid changes

FFT = an efficient algorithm for computing the DFT.

High frequencies contain both useful detail and unwanted noise.

Why noise causes false edges

Derivatives respond to rapid changes. Random pixel fluctuations can therefore create large responses. Smoothing first reduces these fluctuations but can weaken genuine edges.

How filtering connects the views

Spatial convolution corresponds to multiplication in frequency space with compatible boundary handling. Frequency describes variation, not an object’s meaning.

Harris corners: calculate before interpreting

GIVEN

Use equally weighted gradients:
(2, 0), (0, 2), (1, 1).

Build the 2 × 2 structure matrix from sums of gradient products.

$$R = \det(M) - 0.04\,\mathrm{tr}(M)^2$$

YOUR TASK

a. Construct M.

b. Calculate determinant, trace and R.

c. Explain flat, edge and corner cases using eigenvalues.

d. What steps are needed after calculating R?

Try the question before revealing the worked answer.

A8 · Build the structure matrix from products

$$A = \sum I_x^2 = 4 + 0 + 1 = 5$$

$$B = \sum I_x I_y = 0 + 0 + 1 = 1$$

$$C = \sum I_y^2 = 0 + 4 + 1 = 5$$

$$M = \begin{bmatrix} 5 & 1 \\ 1 & 5 \end{bmatrix}$$

Why use gradient products?

Squared gradients measure change in each direction. The cross-product records how horizontal and vertical changes occur together across the local window.

What the matrix summarizes

The matrix predicts how much the patch changes under a small shift. The diagonal entries measure directional strength, while the off-diagonal entries describe coupling.

Construct M from the local gradients before applying the Harris formula.

A8 · A large positive response suggests a corner

DETERMINANT AND TRACE

$$\det(M) = 5(5) - 1(1) = 24, \quad \operatorname{tr}(M) = 10$$

HARRIS RESPONSE

$$R = 24 - 0.04(10)^2 = 20$$

EIGENVALUES IN THIS EXAMPLE

$$\lambda_1 = 6, \qquad \lambda_2 = 4$$

Flat, edge or corner?

Both eigenvalues small: a flat patch. One large and one small: an edge. Both large: a corner, because shifting in any direction changes the patch.

Why R alone is insufficient

Keep responses above a threshold, then apply non-maximum suppression to select local peaks. R depends on gradient magnitude and window scale, so it is not a probability.

Thresholding finds strong candidates; non-maximum suppression localizes them.

Which descriptor match is distinctive?

GIVEN

Descriptors: a = [2, 1], b = [5, 5].
Use Euclidean distance.

Let d_1 and d_2 be the distances to the nearest and second-nearest candidates.

$$\frac{d_1}{d_2} < 0.75$$

YOUR TASK

a. Define keypoint and descriptor. Calculate $\|a - b\|$.

b. Test $(d_1, d_2) = (4, 10)$ and $(4, 4.5)$.

c. Why can an accepted match still be wrong?

d. How does SIFT help with scale and rotation?

Try the question before revealing the worked answer.

A9 · Distance measures similarity; the ratio tests ambiguity

DESCRIPTOR DISTANCE

$$\|a-b\|_2=\sqrt{(2-5)^2+(1-5)^2}=5$$

A CLEARLY BETTER FIRST CANDIDATE

$$\frac{4}{10}=0.40<0.75 \quad \text{accept}$$

TWO NEARLY EQUALLY GOOD CANDIDATES

$$\frac{4}{4.5}\approx 0.889>0.75 \quad \text{reject}$$

Keypoint versus descriptor

A keypoint identifies a distinctive location. A descriptor encodes the surrounding appearance as numbers so candidate patches can be compared.

Why use a ratio?

A small nearest-neighbour distance alone may be ambiguous. The ratio asks whether the best match is substantially better than the runner-up. A lower ratio is stronger evidence.

A distinctive descriptor match is a candidate correspondence, not a guarantee.

A9 · SIFT improves robustness without guaranteeing a match

SIFT: high-level sequence

1. Detect keypoints across scales.

A feature can be found even when its apparent size changes.

2. Assign an orientation.

Describe the patch relative to that orientation to reduce rotation sensitivity.

3. Compare local descriptors.

Why an accepted match can be wrong

Repeated windows, bricks or tiles can look nearly identical. Even the best descriptor match may refer to a different physical location.

What still needs checking

Use geometric consistency to test whether matches support one plausible transformation. Large viewpoint changes, occlusion and weak texture can still break SIFT matching.

For this exam, understand SIFT's purpose and invariances without detailed derivations.

Do the matches agree geometrically?

GIVEN

H = [1 0 0; 0 1 0; 0.02 0 1].

Source point: (50, 30).
Candidate match: (27, 16).
Inlier threshold: 3 pixels.

$$[u, v, s]^T = H[x, y, 1]^T$$

YOUR TASK

a. Apply H and divide by the homogeneous coordinate.

b. Compute reprojection error and classify the match.

c. Explain how RANSAC handles wrong matches.

d. When is a homography suitable, and when can it fail?

Try the question before revealing the worked answer.

A10 · Project first, then measure reprojection error

HOMOGENEOUS COORDINATES

$$u = 50, \quad v = 30, \quad s = 0.02(50) + 1 = 2$$

NORMALIZE TO IMAGE COORDINATES

$$(x', y') = \left(\frac{50}{2}, \frac{30}{2}\right) = (25, 15)$$

COMPARE WITH THE OBSERVED MATCH

$$e = \sqrt{(27 - 25)^2 + (16 - 15)^2} = \sqrt{5} \approx 2.24$$

Why division matters

A homography produces homogeneous coordinates. Dividing the first two components by the third returns the ordinary 2D image position.

Interpret the error

The observed match lies about 2.24 pixels from the prediction. Since this is below 3, it is an inlier for this model and threshold. That does not independently prove it is a true match.

An inlier is a match consistent with a chosen model within a tolerance.

A10 · RANSAC searches for a consistent group

1. Sample and fit

Choose four nondegenerate point pairs and estimate a homography.

2. Count supporting matches

Project points and count errors below the threshold. Repeat with different samples.

3. Keep and refine

Keep a strong consensus and refit using its inliers.

When one homography fits

A planar scene or pure camera rotation can produce a single homography relating the images.

Why the model can fail

Depth variation with camera translation causes parallax. Repeated textures may support a wrong transformation. RANSAC needs enough true matches and a suitable model.

Geometric verification checks agreement across matches, not just patch similarity.

From features to scores to probabilities

GIVEN

x = [1, 2], all biases are zero.

wA = [1, 0]
wB = [0, 1]
wC = [–1, 0]
True class: A.

$$z_k = \mathbf{w}_k^T \mathbf{x} + b_k$$

YOUR TASK

a. Explain features, weights and biases.

b. Calculate all scores and predict the class.

c. Calculate the softmax probabilities.

d. Is the prediction correct? Does confidence guarantee correctness?

Try the question before revealing the worked answer.

A11 · Linear scores rank the classes

$$z_A = 1(1) + 0(2) + 0 = 1$$

$$z_B = 0(1) + 1(2) + 0 = 2$$

$$z_C = -1(1) + 0(2) + 0 = -1$$

$$\hat{y} = \operatorname{argmax}_k z_k = B$$

What each part contributes

Features encode the image. Each weight controls how strongly a feature changes a class score. A bias shifts a score even when the features are zero.

Why B is selected

B has the largest score: 2. Scores need not lie between 0 and 1, and they need not sum to 1. Softmax converts them into normalized probabilities.

The predicted class comes from the largest score, not from the true label.

A11 · Softmax confidence can accompany a wrong prediction

$$p_k = \frac{e^{z_k}}{\sum_j e^{z_j}}$$

$$D = e^{1} + e^{2} + e^{-1} \approx 10.4752$$

Class	Numerator	Probability
A	$e^{1} \approx 2.7183$	0.2595
B	$e^{2} \approx 7.3891$	0.7054
C	$e^{-1} \approx 0.3679$	0.0351

Correctness compares the predicted label with the true label.

Why these are probabilities

Every exponential is positive. Dividing by their sum gives values between 0 and 1 that sum to 1. Softmax preserves the score ranking.

Why confidence is not correctness

The model assigns B about 70.5%, but the true class is A. The prediction is wrong. A high assigned probability does not guarantee truth or calibration.

Regression: calculate MSE, then update a weight

GIVEN

Data: x = [1, 2, 3], y = [3, 5, 7].

Model: $\hat{y} = wx + b$.
Initially w = 1 and b = 1.
For the update, use only x = 3, y = 7.

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

YOUR TASK

a. Calculate predictions and residuals.

b. Calculate MSE and explain squaring.

c. For the third example, calculate $\partial L/\partial w$.

d. Update w with $\eta = 0.05$. Hold b fixed and recalculate that example's loss.

Try the question before revealing the worked answer.

A12 · MSE averages squared prediction errors

x	y	$\hat{y} = x + 1$	$\hat{y} - y$	Error²
1	3	2	–1	1
2	5	3	–2	4
3	7	4	–3	9

$$\text{MSE} = \frac{1+4+9}{3} = \frac{14}{3} \approx 4.667$$

For regression, the output is a numerical value rather than a class label.

What a residual means

The residual is prediction minus target. All three are negative, so the current line underestimates every target in this dataset.

Why square and average?

Squaring prevents sign cancellation and gives large errors more influence. Averaging gives loss per example. MSE has units equal to the target units squared.

A12 · One-example gradient descent moves the line

DIFFERENTIATE THE SINGLE-EXAMPLE LOSS

$$\frac{\partial L}{\partial w} = 2(wx + b - y)x = 2(-3)(3) = -18$$

SUBTRACT THE LEARNING-RATE-SCALED GRADIENT

$$w_{\text{new}} = 1 - 0.05(-18) = 1.9$$

CHECK THE SAME EXAMPLE AGAIN

$$\hat{y}_{\text{new}} = 1.9(3) + 1 = 6.7, \quad L_{\text{new}} = 0.09$$

Why the weight increases

The old prediction is 4 while the target is 7. The gradient is negative, so subtracting it increases w. With positive x, that raises the prediction toward 7.

Which loss improved?

The single-example loss falls from 9 to 0.09. This update uses only the third example and holds b = 1 fixed. It is not the gradient of the three-example mean loss.

Keep the averaging convention and the selected training examples explicit.

Cross-entropy: improve loss without changing the class

GIVEN

Reuse Q11: x = [1, 2], z = [1, 2, −1].
True class: A, $p_A \approx 0.259496$.

Update only wA1 from 1.
Learning rate η = 0.2.

$$L = -\ln p_A, \qquad \frac{\partial L}{\partial w_{A1}} = (p_A - 1)x_1$$

YOUR TASK

a. Calculate cross-entropy.

b. Calculate the gradient and updated weight.

c. Recompute the prediction. Can loss still improve?

d. Explain the complete training and evaluation cycle.

Try the question before revealing the worked answer.

A13 · A low true-class probability produces a high loss

LOSS USES THE TRUE CLASS

$$L = -\ln(0.259496) \approx 1.349012$$

CALCULATE THE GIVEN GRADIENT

$$g = (0.259496 - 1)(1) = -0.740504$$

UPDATE ONLY THE SPECIFIED WEIGHT

$$w_{A1}^{\text{new}} = 1 - 0.2(-0.740504) = 1.148101$$

Why use the probability of A?

The target is A. Cross-entropy penalizes the model for assigning A little probability, regardless of which class currently wins. Higher true-class probability means lower loss.

How the update helps

A negative gradient makes wA1 increase. Since $x_1 = 1$, this raises A's score. Other weights and biases remain fixed for this exercise.

Use the true label to select the loss, not the model's predicted label.

A13 · Probability can improve before the decision changes

$$\mathbf{z}_{\text{new}} = [1.148101,\ 2,\ -1]$$

$$p_A^{\text{new}} = \frac{e^{1.148101}}{e^{1.148101} + e^{2} + e^{-1}} \approx 0.288951$$

$$L_{\text{new}} = -\ln(0.288951) \approx 1.2415$$

Predicted class: B before and after.

Loss is more sensitive than accuracy

A gains probability, so its loss decreases. However, B's score is still largest. Accuracy changes only when the winning class changes.

The training cycle

Predict, calculate loss, differentiate, then update parameters. Repeat on training data. Use validation results for model selection and reserve test data for final evaluation.

A lower loss can reflect progress even while the classification error stays unchanged.

How large should a gradient-descent step be?

GIVEN

Use this simple quadratic loss.
Start at w = 3, so L = 4.5.

Each learning-rate experiment starts again at w = 3.

$$L(w) = \frac{1}{2}w^2, \qquad \frac{dL}{dw} = w$$

YOUR TASK

a. Update once with η = 0.2 and find the loss.

b. Repeat with η = 0.8.

c. Repeat with η = 2.4. Explain the result.

d. What happens with tiny steps? Does crossing zero always mean failure?

Try the question before revealing the worked answer.

A14 · The learning rate changes the step size

$$w_{\text{new}} = w - \eta w = (1 - \eta)w$$

η	Updated w	New loss
0.2	3 – 0.2(3) = 2.4	½(2.4)² = 2.88
0.8	3 – 0.8(3) = 0.6	½(0.6)² = 0.18
2.4	3 – 2.4(3) = –4.2	½(–4.2)² = 8.82

Compare with the initial loss: 4.5.

A larger step can be faster, but a step that is too large can increase loss.

Moderate steps reduce this loss

Both 0.2 and 0.8 move closer to the minimum at w = 0. Here 0.8 gives the bigger improvement in one step.

A large step overshoots

With 2.4, the update crosses the minimum and lands farther away. The loss increases to 8.82. Repeating this step multiplies the weight by –1.4 and diverges.

A14 · Crossing the minimum can still converge

REPEATED UPDATES FOR THIS QUADRATIC

$$w_{t+1} = (1 - \eta) w_t$$

DISTANCE TO ZERO SHRINKS WHEN

$$|1 - \eta| < 1 \quad \Longleftrightarrow \quad 0 < \eta < 2$$

AN OSCILLATING BUT CONVERGENT EXAMPLE

$$\eta = 1.5: \quad 3, \ -1.5, \ 0.75, \ -0.375, \ldots$$

Why tiny steps can be slow

If η is very small, the multiplier 1 – η is close to 1. Each update removes only a small fraction of the error, requiring many steps.

Why crossing zero is not failure

For 1 < η < 2, the sign alternates but the magnitude shrinks. The range 0 < η < 2 applies to this particular loss; real models depend on their loss curvature.

Judge the loss and distance to the minimum, not just the sign of the weight.

Which checkpoint generalizes best?

GIVEN

Synthetic training / validation losses:

Epoch 1: 0.80 / 0.90
Epoch 2: 0.50 / 0.60
Epoch 3: 0.30 / 0.45
Epoch 4: 0.15 / 0.55

YOUR TASK

a. Explain training, validation and test roles.

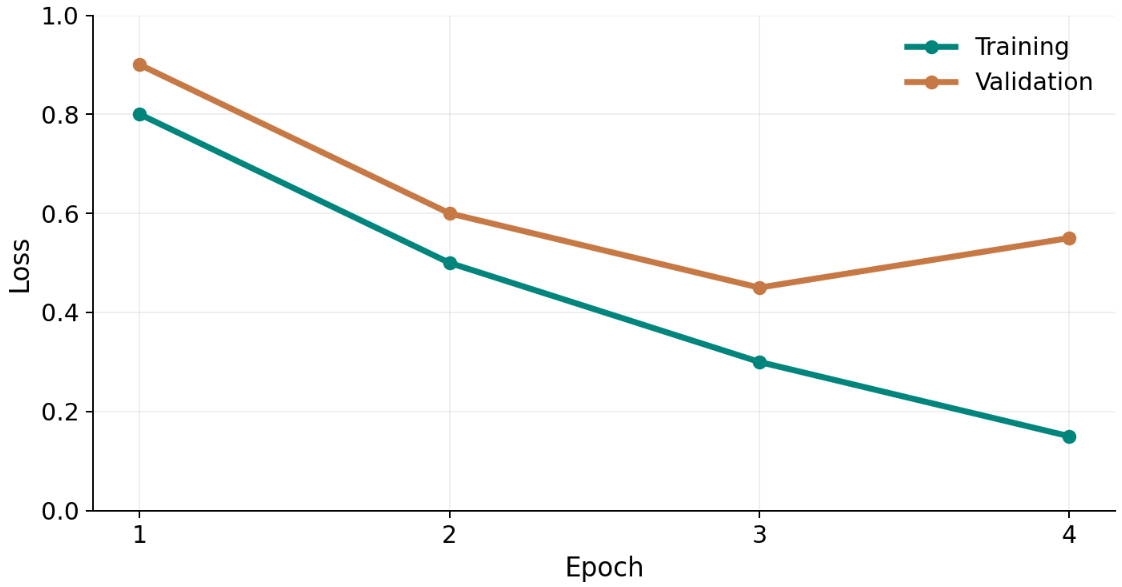
b. Select a checkpoint using validation loss.

c. Interpret epochs 3 to 4. What would underfitting look like?

d. Why avoid test tuning and nearly identical frames across splits?

Try the question before revealing the worked answer.

A15 · Validation loss selects epoch 3



Lowest validation loss: 0.45 at epoch 3.

The selection rule

If validation loss is the chosen metric, keep the epoch-3 checkpoint. Training loss alone would favour epoch 4, but it measures fit to examples already used for learning.

The overfitting warning

From epoch 3 to 4, training improves while validation worsens. The model may be fitting training-specific details. A single fluctuation is a warning, not proof by itself.

Model selection should reflect performance on data outside parameter fitting.

A15 · Separate fitting, selection and final evaluation

Split	Purpose	Typical use
Training	Fit parameters	Gradient updates
Validation	Select the model	Learning rate, checkpoint
Test	Final evaluation	Report held-out performance

Underfitting: both sets perform poorly.

A model may be too simple, use weak features, or need better optimization. More training is not always sufficient.

Why test-driven tuning is biased

Repeatedly choosing settings from test results adapts the model to that test set. The final score then becomes an optimistic estimate of unseen performance.

Why independent splits matter

Near-identical video frames can leak across splits. When appropriate, split by video or subject. Generalization means performing well on unseen data from the intended task.

Keep test data out of model selection, and keep strongly related examples together.

A two-class classifier: sigmoid, MSE and one update

GIVEN

Class 0: normal image.
Class 1: defective image.

$x = [2, 1]$, true label $y = 1$.
$w_1 = w_2 = b = 0$, learning rate $\eta = 0.1$.
Use one-example MSE without a ½ factor.

$$z = w_1 x_1 + w_2 x_2 + b, \quad p = \frac{1}{1 + e^{-z}}$$

YOUR TASK

a. Find z, p, the predicted class and MSE. Use class 1 if $p \geq 0.5$.

b. Use the chain rule for both weight gradients and the bias gradient.

c. Update w_1, w_2 and b simultaneously.

d. Recalculate p and MSE. Explain the improvement.

Try the question before revealing the worked answer.

A16 · Step 1: predict a probability, then measure error

LINEAR SCORE

$$z = 0(2) + 0(1) + 0 = 0$$

SIGMOID PROBABILITY

$$p = \sigma(z) = \frac{1}{1 + e^{-0}} = \frac{1}{2} = 0.5$$

ONE-EXAMPLE MSE

$$L = (p - y)^2 = (0.5 - 1)^2 = 0.25$$

How one output describes two classes

p is the probability of class 1. Class 0 has probability 1 – p. Initially both equal 0.5, placing this example on the decision boundary.

Why a correct label still has loss

The stated tie rule selects class 1, matching y = 1. But p = 0.5 is far from the target probability 1, so MSE remains positive. Training uses p, not the thresholded label.

Keep the score z, probability p, predicted label and true label conceptually separate.

A16 · Step 2: differentiate MSE and sigmoid

LOSS DERIVATIVE

$$\frac{\partial L}{\partial p} = 2(p - y) = 2(0.5 - 1) = -1$$

SIGMOID DERIVATIVE

$$\frac{\partial p}{\partial z} = \frac{e^{-z}}{(1 + e^{-z})^2} = p(1 - p)$$

EVALUATE AT THE CURRENT PROBABILITY

$$\frac{\partial p}{\partial z} = 0.5(1 - 0.5) = 0.25$$

For sigmoid with MSE, do not omit the factor p(1 – p).

Where the factor of 2 comes from

Differentiating the square gives 2(p – y). This exercise uses L = (p – y)². A loss with an extra ½ would cancel the factor of 2.

Why sigmoid needs its own derivative

The weights change z, and z changes p. The loss depends on the weights through that intermediate probability, so the sigmoid derivative must enter the chain rule.

A16 · Step 3: carry the gradient back to each parameter

CHAIN RULE FOR THE SCORE

$$\delta = \frac{\partial L}{\partial z} = \frac{\partial L}{\partial p}\frac{\partial p}{\partial z} = (-1)(0.25) = -0.25$$

WEIGHT GRADIENTS

$$\frac{\partial L}{\partial w_j} = \delta \frac{\partial z}{\partial w_j} = \delta x_j$$

BIAS GRADIENT

$$\frac{\partial L}{\partial b} = \delta \frac{\partial z}{\partial b} = \delta(1) = -0.25$$

Substitute each feature value

For w_1, $x_1 = 2$, giving $-0.25 \times 2 = -0.50$.
For w_2, $x_2 = 1$, giving $-0.25 \times 1 = -0.25$.

Why the bias behaves differently

Changing b changes z directly by the same amount, so its derivative is 1. All gradients come from the same old weights and bias before any update.

The feature value scales its weight gradient; the bias has no feature multiplier.

A16 · Step 4: subtract the gradient from each parameter

FIRST WEIGHT

$$w_1^{\text{new}} = 0 - 0.1(-0.50) = 0.05$$

SECOND WEIGHT

$$w_2^{\text{new}} = 0 - 0.1(-0.25) = 0.025$$

BIAS

$$b^{\text{new}} = 0 - 0.1(-0.25) = 0.025$$

The new model is $z = 0.05x_1 + 0.025x_2 + 0.025$.

The update rule

For any parameter θ, use θnew = θold – η ∂L/∂θ. Subtracting a negative gradient increases the parameter, moving locally toward lower loss.

What simultaneous means

Calculate every gradient using the original state, then update all three values. Recomputing a later gradient after changing a weight would be a different update procedure.

A16 · Step 5: verify the update

NEW SCORE

$$z_{\text{new}} = 0.05(2) + 0.025(1) + 0.025 = 0.15$$

NEW PROBABILITY

$$p_{\text{new}} = \frac{1}{1 + e^{-0.15}} \approx 0.537430$$

NEW MSE

$$L_{\text{new}} = (0.537430 - 1)^2 \approx 0.213971$$

What improved

The probability of the true class rises from 0.5 to about 0.5374. MSE decreases from 0.25 to about 0.214. The prediction stays class 1, so accuracy is unchanged.

What this does not establish

This is improvement on one training example. Assess generalization on separate data. For a batch, average the example gradients before taking the update.

A useful update can reduce loss even when no predicted label changes.