CS455 Computer Vision: Practice Final Examination

University of Washington
2026

Solution:

SOLUTIONS

Name: _____

Student ID: _____

Question	Total	**Points**
True/False	18	
Multiple Choice	30	
Short answers 1: Camera Projection	16	
Short answers 2: Seam Carving	8	
Short answers 3: Dimensionality reduction	15	
Short answers 4: Videos	11	
Short answers 5: Segmentation	15	
Short answers 6: Linear Classification	14	
Short answers 7: Backpropagation	14	
Short answers 8: Deep Learning	14	
Extra credits in short answers	16	
Total	100	

Instructions:

1. This practice final examination contains a set of exemplary problems. *The final examination will be significantly shorter than this.* Expect 10-15 true/false, 10-15 multiple choice, and 4-7 short answer questions.

2. In the final examination, you have **one hour fifty minutes** to complete the examination. As a courtesy to your classmates, we ask that you not leave during the last fifteen minutes.

3. For **True/False** and **Multiple-Choice** sections, correct answers will get the points listed next to the question. You will receive 0 for all unanswered questions and -0.5 for all incorrect answer. You can choose to not answer questions that you are unsure of to avoid receiving a negative point.

4. For **Short Answers**, you will NOT get negative points for incorrect answers. Partial credit will be assigned for showing your work and reasoning.

5. You may use one double-sided 8.5" $\times$ 11" **hand-written** sheet with notes that you have prepared. You may not use any other resources, including lecture notes, books, other students or other engineers. These notes must be submitted along with your booklet.

6. Please sign the below Honor Code statement.

In recognition of and in the spirit of the University of Washington Honor Code, I certify that I will neither give nor receive unpermitted aid on this examination.

Signature: _____

True/False (18 points)

Fill in the circle next to True or False, or fill in neither. Fill it in completely.
No explanations are required.

1. Using the forward seam algorithm allows us to determine the seam that minimizes the amount of energy inserted to the image.
   - √ **True**
   - ◯ False
2. Cross-correlation is a non-linear operation because it involves shifting kernels.
   - ◯ True
   - √ **False**
3. Histogram of Oriented Gradients (HoG) descriptors are invariant to image rotation, making them suitable for object detection under varying viewpoints.
   - ◯ True
   - √ **False**
4. Corners generally have high gradients in a single direction.
   - ◯ True
   - √ **False**
5. Harris corners are scale-invariant by default
   - ◯ True
   - √ **False**
6. The output size of an image after convolution is always the same as the input image.
   - ◯ True
   - √ **False**
7. A projective transformation always preserves straight lines but may change the relative distances between points on those lines.
   - √ **True**
   - ◯ False
8. Principal Component Analysis (PCA) can be used for image compression by reducing the dimensionality of image data.
   - √ **True**
   - ◯ False
9. When using Hough Transform to detect **lines**, we voted within **2D** cells to find the most likely lines in an image. Recall that for detecting **lines** we represented a line as $x \cdot cos(\theta) + y \cdot sin(\theta) = \rho$. If we were to use Hough Transform to detect **circles**, we would also vote in 2D cells.
   - ◯ True
   - √ **False**
10. The optical flow algorithm works better in videos with lower frame rates.
    - ◯ True
    - √ **False**

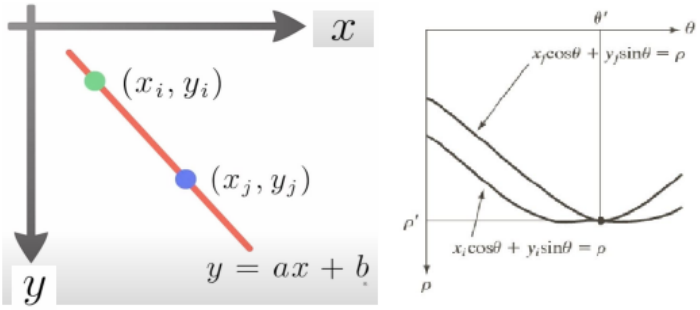


Figure 1: Hough Transform on Lines

11. Increasing the size of the integration window in Lucas–Kanade improves accuracy at the cost of computation time.

    ◯ True

    √ **False**

12. When we use linear classifiers, we will get the probability of an image input belonging to a class.

    ◯ True

    √ **False**

13. Increasing the number of clusters in k-means generally decreases the sum of squared distances within clusters.

    √ **True**

    ◯ False

14. The centroid of a cluster in k-means is always one of the data points in the dataset.

    ◯ True

    √ **False**

15. Moving average filter has superposition property but it is not causal and stable.

    ◯ True

    √ **False**

16. Image segmentation filter is additive but not homogeneous.

    ◯ True

    √ **False**

17. Convolution and correlation operations yield to the same results using kernel $K = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$

    √ **True**

    ◯ False

18. Convolution and cross-correlation are identical if the kernel is symmetric around its center.

    √ **True**

    ◯ False

Multiple Choice (30 points)

1. **Harris Corners (1 point).** Which of the following transformations would **break** the assumption underlying the Harris detector? (*Choose all that apply*):

   ☐ Image rotation

   ☐ Uniform brightness shift

   √ **Extreme affine shear**

   √ **Non-uniform scaling**

2. **HoG Descriptor (1 point).** Which of the following makes HoG descriptors effective for object detection? (*Choose the correct answer*):

   ◯ They rely on color histograms for feature extraction

   √ **They normalize gradient orientations within local image blocks**

   ◯ They use keypoint matching for spatial alignment

   ◯ They are computed using a single global histogram

3. **SIFT (2 points).** SIFT descriptors are always invariant to which of the following? (*Choose all that apply*):

   √ **Translation**

   ☐ Affine transformations

   √ **Rotation**

   ☐ Occlusion

   √ **Scale**

4. **Cameras (2 points).** What is the role of the intrinsic matrix in a camera model? (*Choose the correct answer*):

   ◯ It maps 2D image coordinates to 3D world coordinates

   ◯ It describes the camera's position and orientation in the world.

   √ **It maps 3D camera coordinates to 2D image coordinates.**

   ◯ It provides the camera's field of view and resolution.

5. **Cameras (2 points).** Given a camera with intrinsic matrix K (3×3) and extrinsic parameters $[R \mid t]$ where R is 3×3 and t is 3×1, how would you obtain the 2D image coordinates P_{2D} of a 3D world point P_w (3×1)? (Hint: Verify that the matrix dimensions are consistent.)

   (*Choose the correct answer*):

   ◯ $P_{2D} = K \cdot P_w$

   ◯ $P_{2D} = R \cdot P_w + t$

   √ $P_{2D} = K \cdot (R \cdot P_w + t)$

   ◯ $P_{2D} = (R \cdot P_w + t) \cdot K$

6. **Perspective Projection (1 point).** Which of the following are **invariant** under perspective projection? (*Choose all that apply*):

   √ **Straight lines remain straight**

   ☐ Parallel lines remain parallel

   √ **Ratios of distances along a line are preserved**

   ☐ Angles between lines are preserved

7. **k-means (2 point).** How is the optimal number of clusters typically determined in k-means clustering (*Choose the correct answer*)?

   ◯ By selecting the number of clusters that minimizes the within-cluster variance

   ◯ By selecting the number of clusters randomly

   ◯ By choosing a prime number of clusters

   √ **By using cross-validation**

8. **SIFT (2 points).** In the SIFT descriptor, what is the size of the descriptor vector generated for each keypoint? (*Choose the correct answer*):

   ◯ 32

   ◯ 64

   √ **128**

   ◯ 256

   Explanation: The descriptor is computed using 16 histograms with 8 orientation bins each, resulting in $16 \times 8 = 128$ values.

9. **SIFT (2 points).** If two keypoints have similar orientation histograms but one has significantly higher gradient magnitudes, how will this affect their SIFT descriptors? (*Choose the correct answer*):

   ◯ The keypoint with higher gradient magnitudes will dominate the descriptor comparison, making it more likely to match.

   ◯ The descriptors will be identical since orientation, not magnitude, defines the descriptor.

   √ **The descriptors will remain comparable because SIFT normalizes the gradient magnitudes.**

   ◯ The keypoint with lower gradient magnitudes will be discarded during descriptor computation.

10. **Convolutions (2 points).** Which of the following affects the output size of convolution operations? (*Choose all that apply*):

    √ **Kernel Size**

    √ **Padding**

    √ **Stride**

    ☐ Activation function

11. **Filters (2 points).** A moving average filter is an example of (*Choose the correct answer*):

    ◯ A system that amplifies high-frequency components

    √ **A linear, shift-invariant system used for smoothing**

    ◯ A filter that preserves sharp edges in an image

    ◯ A non-causal system that depends on future inputs

12. **Linearity (1 point).** A system is considered linear if it satisfies (*Choose all that apply*):

    √ **Additivity**

    ☐ Shift invariance

    √ **Homogeneity**

    √ **Superposition**

    ☐ Stability

13. **Deep Learning (1 points).** Which of the following statements about Convolutional Neural Networks (CNNs) is TRUE? (*Choose the correct answer*):

◯ Fully connected layers are applied to entire images without any spatial structure.

◯ Pooling layers increase the spatial resolution of feature maps.

√ **CNNs use weight sharing to reduce the number of parameters.**

◯ CNNs do not use non-linearity functions such as ReLU.

14. **Edges (2 points).** What is the primary goal of non-maximum suppression in edge detection algorithms like Canny? (*Choose the correct answer*):

◯ To blur the edges in an image and avoid thin edges.

√ **To produce precise edges by removing weaker edge pixels.**

◯ To highlight edges belonging to object boundaries.

◯ To noisy edges from the corners of the image.

15. **PCA (1 point).** If you apply PCA to a 2D dataset where all points lie perfectly on a single line, what would you expect the eigenvalue associated with the second principal component to be? (*Choose the correct answer*):

◯ Equal to the eigenvalue of the first principal component

√ **Approximately zero**

◯ Negative

◯ Larger than the eigenvalue of the first principal component

16. **Panoramas (2 points).** In image stitching (panorama creation), which method is commonly used to remove outliers while estimating the affine transformation matrix? (*Choose the correct answer*):

◯ Principal Component Analysis (PCA).

◯ Singular Value Decomposition (SVD).

√ **Random Sample Consensus (RANSAC).**

◯ K-Means Clustering.

17. **Optical Flow (2 points).** Which assumption is fundamental to most optical flow algorithms? (*Choose the correct answer*):

◯ Scale invariance

√ **Brightness constancy**

◯ Rotation invariance

◯ Temporal discontinuity

18. **Detection (2 points).** Which of the following statements about Intersection over Union (IoU) are true? (*Choose all that apply*):

☐ A higher IoU threshold will result in a higher true positive rate.

√ **IoU is used to determine if a detected object matches the ground truth.**

☐ A higher IoU threshold always improves object detection performance.

√ **The IoU threshold is a hyperparameter that affects the precision-recall trade-off.**

Short Answers 1: Camera Projection and Transformations (16 points)

1. **(16 points)** A calibrated camera with the intrinsic matrix:

$$K = \begin{bmatrix} 1000 & 0 & 320 \\ 0 & 1000 & 240 \\ 0 & 0 & 1 \end{bmatrix}$$

is positioned at $(3, 2, 10)$ in world coordinates with its principal axis aligned with the Z-axis.

(a) **(4 points)** A 3D point $(X_w, Y_w, Z_w) = (6, 4, 15)$ is observed. Compute its camera coordinates (X_c, Y_c, Z_c).

Solution: $X_c = X_w - C = (6, 5, 15) - (3, 2, 10) = (3, 2, 5)$

(b) **(4 points)** Using the intrinsic matrix K, compute the image coordinates (u, v) of the projected 2D point corresponding to (X_c, Y_c, Z_c).

Solution: $K \cdot X_c = \begin{bmatrix} 100 & 0 & 320 \\ 0 & 1000 & 240 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 3 \\ 2 \\ 5 \end{bmatrix} = \begin{bmatrix} 4600 \\ 3200 \\ 5 \end{bmatrix}$

$$u = \frac{4600}{5} = 920$$

$$v = \frac{3200}{5} = 640$$

(c) **(4 points)** What effect does increasing the focal length f by a factor of 2 have on the image projection? Explain mathematically.

Solution: Recall from part b how we obtained u and v.

$$u = f \cdot \frac{x_c}{z_c} + c_x \rightarrow u - c_x = f \cdot \frac{x_c}{z_c}$$

$$v = f \cdot \frac{y_c}{z_c} + c_y \rightarrow v - c_y = f \cdot \frac{y_c}{z_c}$$

Now if we doubled the focal length, we would get that

$$u' - c_x = \frac{2fx_c}{z_c}$$

$$v' - c_y = \frac{2fy_c}{z_c}$$

We can see from this that the distance of our projected point (u', v') from (c_x, c_y) has doubled, effectively zooming into the image.

(d) **(4 points)** If the camera is tilted downward by 30 degrees along the X-axis, derive the new rotation matrix and compute the transformed camera coordinates.

Solution:

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{bmatrix}$$

We're given that the camera is tilted downwards along the X-axis. Thus this rotation along the X-axis is $\theta = -30$ so we plug it in to get:

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{\sqrt{3}}{2} & \frac{1}{2} \\ 0 & -\frac{1}{2} & \frac{\sqrt{3}}{2} \end{bmatrix}$$

We can now use this to compute the transformed camera coordinates $X_c = R_x \cdot \begin{bmatrix} 3 \\ 2 \\ 5 \end{bmatrix} = \begin{bmatrix} 3 \\ \sqrt{3} + 2.5 \\ -1 + \frac{5\sqrt{3}}{2} \end{bmatrix}$

Short Answers 2: Seam Carving (8 points)

2. **(8 points)**

(a) **(4 points)** What is the cost matrix equation for cell $M(i, j)$, assuming we're doing seam carving on vertical seams and $i \neq 0$, in the following cases. You can denote the energy function as $E(i, j)$.

(i) **(2 points)** Seam carving:

Solution: $M(i,j) = E(i,j) + min \begin{cases} M(i-1, j-1) \\ M(i-1, j) \\ M(i-1, j+1) \end{cases}$

(ii) **(2 points)** Forward seam carving:

Solution: $M(i,j) = E(i,j) + min \begin{cases} M(i-1, j-1) + C_{left} \\ M(i-1, j) + C_{middle} \\ M(i-1, j+1) + C_{right} \end{cases}$
where $C_{left} = |I(i, j+1) - I(i, j-1) + |I(i-1, j) - I(i, j-1)|$
$C_{middle} = |I(i, j+1) - I(i, j-1)|$
$C_{right} = |I(i, j+1) - I(i, j-1)| + |I(i-1, j) - I(i, j+1)|$

(b) **(4 points)** Let's say we have a picture, but it has an unwanted object in the background. Describe the steps you would take to remove the object from the background, but still preserve the size of the image, using seam carving.

Solution: We want to compute the energy function $E(j, i)$ while manually setting very negative values for the pixels in the unwanted object. Perform seam carving or forward seam carving to iteratively remove seams. Once the unwanted object is removed, we can restore the width of the image by inserting low energy seams.

Short Answers 3: PCA and LDA (15 points + 5 extra credits.)

3. **PCA (8 Points)**

(a) **(3 points.)** Explain in words what projecting points onto the first principal component aims to do.

Solution: Projecting points onto the first principal component aims to lower the dimensionality of the points to 1 while maximizing the variance of the points in the resulting projection

(b) **(5 points.)** Suppose your image is represented by a 2D point $X = (5, 2)$ and you've computed the first principal component of your dataset is the vector (2, 2). Project the point in X onto the first principal component. The formula for projecting a point p onto a vector v is $\frac{p \cdot v}{||v||^2} v$

Solution: First the dot product is 14 and the norm squared of v is 8. Dividing gives $\frac{7}{4}$. Multiply that by v and we get $[\frac{7}{2}, \frac{7}{2}]$

4. **LDA (7 Points)**

(a) **(3 points.)** Explain the main difference between PCA and LDA in terms of their objectives.

Solution: The key difference is supervised vs unsupervised decomposition. PCA focuses on the variance withing the data itself, irrespective of any class labels. LDA focuses on maximizing the sepration between multiple known classes.

(b) **(4 points.)** Which is generally better when dealing with a classification problems in computer vision?

Solution: LDA. The main advantage of using Linear Discriminant Analysis (LDA) over Principal Component Analysis (PCA) when dealing with classification problems in computer vision—or indeed any classification problem—stems from LDA's focus on maximizing class separability. LDA ensures that classes are better separated on the most relevant features for classification.

(c) **(Extra credit 5 points)** Suppose you have a set of 4 images, where each is represented by two values $X_t \in \mathbb{R}^2$

$$X = \begin{bmatrix} 5 & 3 & 6 & 2 \\ 2 & 4 & 5 & 1 \end{bmatrix}$$

Compute the unit length principal components, and say which one would be used if we are using the first principal component. Hint: Find the eigenvalues/eigenvectors of the covariance matrix $C = (X - \mu)(X - \mu)^T$.

Solution: First we calculate the mean $\mu = [4, 3]$. Then we center X giving $\begin{bmatrix} 1 & -1 & 2 & -2 \\ -1 & 1 & 2 & -2 \end{bmatrix}$

Then $X^T X = \begin{bmatrix} 10 & 6 \\ 6 & 10 \end{bmatrix}$

It's eigenvectors are $[1/\sqrt{2}, 1/\sqrt{2}]$ with eigenvalue 16 and $[1/\sqrt{2}, -1/\sqrt{2}]$ with eigenvalue 4. The first eigenvector is chosen as the eigenvalue is the largest.

Short Answers 4: Videos (11 points).

5. **Optical Flow (10 points)** When calculating optical flow, we made the assumption that the brightness of two consecutive frames does not differ by much. This brightness constancy equation is written as:

$$I_x u + I_y v + I_t = 0$$

(a) **(5 points.)** Explain briefly what each of the five variables in the brightness constancy equation are.

(b) **(2 points.)** According to this equation, explain what is the direction in the image along which optical flow cannot be reliably estimated?

(c) **(3 points.)** List three key assumptions when estimating optical flow.

Short Answers 5: Segmentation (15 points + 5 extra credits.)

You are given a grayscale image of size 256×256 pixels. Your feature space is the grayscale values without any position information. The pixel intensities range from 0 to 255. You are tasked with segmenting this image using the K-means clustering algorithm.

(a) **(2 points)** How does k-means algorithm work in the context of image segmentation? (2 sentences max)

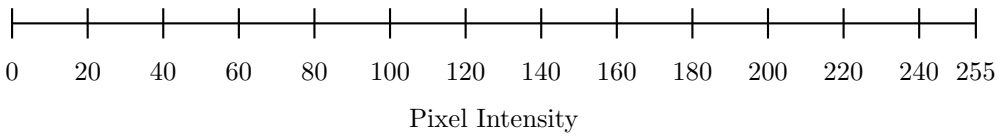
Solution: In image segmentation, K-means clusters pixel intensities (or features), grouping similar intensity values (or features) together to form distinct regions in the image.

(b) **(3 points)** Suppose you are using k-means with $K = 3$. Describe the steps you would take to segment the image. List out the steps of the k-means algorithm.

Solution: 1) Initialize 3 random cluster centroids from the pixel internsities (features). 2) Assign each pixel to the nearest cluster centroid based on its intensity value (feature vector). 3) Update the cluster centroids by computing the mean intensity value of the pixels in each cluster. 4) Repeat steps 2 and 3 until the centroids do not change or reach maximum number of iterations.

(c) **(2 points)** After running K-means clustering on the image, you obtained three clusters with centroids at 40, 120, and 200. How would you assign the pixel values in the segmented image based on these centroids? Show the range of numbers within each cluster on the number line below. Draw two vertical

lines to separate the number line into 3 clusters.



Solution: 40: 0 - 80, 120: 80 - 160, 200: 160 - 255

6. **(8 points.)** Assume $K = 3$ and and the initial centroids are 40, 120, and 200. Given the following five pixel intensities: 30, 58, 100, 180, and 220, perform the assignment and update steps of the K-means algorithm for two iterations and show the results step by step.

Solution: First Iteration:

Assignment: cluster 1: 30, 58 cluster 2: 100 cluster 3: 180, 220

Update: New centroid for Cluster 1: $(30 + 58)/2 = 44$ New centroid for Cluster 2: 100 New centroid for Cluster 3: $(180 + 220)/2 = 200$

Second Iteration:

Assignment: cluster 1: 30, 58 cluster 2: 100 cluster 3: 180, 220

Update: New centroid for Cluster 1: $(30 + 58)/2 = 44$ New centroid for Cluster 2: 100 New centroid for Cluster 3: $(180 + 220)/2 = 200$

7. **(Extra Credit: 5 points.)** The LAB color space is a color-opponent space with three axes: L for lightness, a for the green-red component, and b for the blue-yellow component. Unlike RGB, which is based on the primary colors of light, LAB is designed to approximate human vision and is more perceptually uniform. This means that the same amount of numerical change in these values corresponds to about the same amount of visually perceived change.
   Compare the RGB and LAB color spaces in the context of image segmentation using K-means clustering. Which color space would you recommend and why? (4 sentence max)

   Solution: RGB directly represents colors using red, green, and blue channels, which is simple and intuitive but may not accurately reflect perceptual differences between colors. LAB separates luminance from color information, making it more perceptually uniform and better suited for image segmentation. LAB is recommended because it aligns more closely with human vision, providing better segmentation results by separating intensity from color information.

Short Answers 6: Linear Classification (14 points)

8. **(14 points.)** Consider a linear classifier for a 3-class classification problem. The classifier computes scores using:

$$\mathbf{s} = W\mathbf{x} + \mathbf{b}$$

where $\mathbf{x} \in \mathbb{R}^4$ is the input feature vector, $W \in \mathbb{R}^{3\times 4}$ is the weight matrix, and $\mathbf{b} \in \mathbb{R}^3$ is the bias vector. You are given the following:

$$W = \begin{bmatrix} 1 & 0 & -1 & 2 \\ 0 & 1 & 1 & -1 \\ -1 & 2 & 0 & 1 \end{bmatrix}, \qquad \mathbf{b} = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}, \qquad \mathbf{x} = \begin{bmatrix} 1 \\ 2 \\ 1 \\ 0 \end{bmatrix}$$

The ground-truth label is class 0 (zero-indexed).

(a) **(2 points)** Compute the score vector $\mathbf{s} = W\mathbf{x} + \mathbf{b}$.

Solution:

$$W\mathbf{x} = \begin{bmatrix} (1)(1) + (0)(2) + (-1)(1) + (2)(0) \\ (0)(1) + (1)(2) + (1)(1) + (-1)(0) \\ (-1)(1) + (2)(2) + (0)(1) + (1)(0) \end{bmatrix} = \begin{bmatrix} 0 \\ 3 \\ 3 \end{bmatrix}$$

$$\mathbf{s} = W\mathbf{x} + \mathbf{b} = \begin{bmatrix} 0 \\ 3 \\ 3 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} = \begin{bmatrix} 1 \\ 3 \\ 2 \end{bmatrix}$$

(b) **(2 points)** Using the scores from part (a), compute the softmax probabilities $\mathbf{p} = \text{softmax}(\mathbf{s})$ for each class. You may leave your answer in terms of exact fractions or round to 3 decimal places.

Solution:

$$e^{s_0} = e^1 \approx 2.718, \qquad e^{s_1} = e^3 \approx 20.086, \qquad e^{s_2} = e^2 \approx 7.389$$

$$\sum_k e^{s_k} \approx 2.718 + 20.086 + 7.389 = 30.193$$

$$\mathbf{p} \approx \begin{bmatrix} 2.718/30.193 \\ 20.086/30.193 \\ 7.389/30.193 \end{bmatrix} \approx \begin{bmatrix} 0.090 \\ 0.665 \\ 0.245 \end{bmatrix}$$

(c) **(2 points)** Compute the cross-entropy loss for this example given the ground-truth label is class 0. Based on the loss value and the predicted probabilities, does the classifier appear to be performing well on this example? Briefly justify your answer.

Solution:

$$\mathcal{L} = -\log(p_0) \approx -\log(0.090) \approx \mathbf{2.408}$$

The classifier is performing poorly on this example. It assigns only 9% probability to the correct class (class 0) while assigning 66.5% to the incorrect class 1. A well-performing classifier would assign a probability close to 1.0 to class 0, yielding a loss close to 0.

(d) **(4 points)** Instead of cross-entropy loss, suppose we use the **multiclass SVM (hinge) loss**. For a correct class y and margin $\Delta = 1$, the SVM loss for a single example is:

$$\mathcal{L}_{\text{SVM}} = \sum_{j \neq y} \max(0,\ s_j - s_y + \Delta)$$

(i) Using the scores from part (a) and ground-truth label $y = 0$, compute $\mathcal{L}_{\text{SVM}}$. **(2 points)**

(ii) Describe one key difference in how the SVM loss and cross-entropy loss treat examples that are already correctly classified with a large margin. **(2 points)**

Solution: (i)

$$\mathcal{L}_{\text{SVM}} = \max(0,\ s_1 - s_0 + 1) + \max(0,\ s_2 - s_0 + 1)$$

$$= \max(0,\ 3 - 1 + 1) + \max(0,\ 2 - 1 + 1) = \max(0,\ 3) + \max(0,\ 2) = 3 + 2 = \mathbf{5}$$

(ii) The SVM loss is *saturating*: once the correct class score exceeds all incorrect class scores by at least the margin Δ, the loss is exactly zero and the classifier receives no gradient signal for that example. Cross-entropy loss, by contrast, is *non-saturating*: it continues to push the probability of the correct class toward 1 even when the example is already correctly classified, always producing a non-zero gradient as long as $p_y < 1$. This means cross-entropy continues to refine the decision boundary even for easy examples, while SVM focuses only on examples near or within the margin.

(e) **(4 points)** Suppose we add an $L2$ regularization term to the cross-entropy loss:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{CE}} + \frac{\lambda}{2}\|W\|_F^2$$

where $\|W\|_F^2 = \sum_{i,j} W_{ij}^2$ is the squared Frobenius norm of W.

(ii) Compute $\|W\|_F^2$ for the weight matrix given above. **(1 point)**

(ii) Write out the gradient of $\mathcal{L}_{\text{total}}$ with respect to W, and describe how $L2$ regularization changes the weight update rule under gradient descent. **(2 points)**

(iii) Explain intuitively what $L2$ regularization encourages the classifier to learn, and why this might improve generalization. **(1 point)**

Solution: (i)

$\|W\|_F^2 = 1^2 + 0^2 + (-1)^2 + 2^2 + 0^2 + 1^2 + 1^2 + (-1)^2 + (-1)^2 + 2^2 + 0^2 + 1^2 = 6 + 3 + 6 = \mathbf{15}$

(ii) The gradient of the total loss with respect to W is:

$$\frac{\partial \mathcal{L}_{\text{total}}}{\partial W} = \frac{\partial \mathcal{L}_{\text{CE}}}{\partial W} + \lambda W$$

The weight update rule becomes:

$$W \leftarrow W - \eta\left(\frac{\partial \mathcal{L}_{\text{CE}}}{\partial W} + \lambda W\right) = (1 - \eta\lambda)W - \eta\frac{\partial \mathcal{L}_{\text{CE}}}{\partial W}$$

The $(1 - \eta\lambda)$ factor *shrinks* the weights at every step (sometimes called **weight decay**), in addition to the usual gradient step.

(iii) $L2$ regularization encourages the classifier to spread weight magnitude evenly across all input features rather than concentrating it on a few. This discourages the model from relying too heavily on any single feature, which tends to reduce overfitting and improve generalization to unseen examples.

Short Answers: Backpropagation (14 points)

9. **(14 points.)** Consider a small two-layer neural network with the following architecture. The network takes a scalar input x and produces a scalar output $\hat{y}$:

$$a_1 = \text{ReLU}(w_1 x + b_1), \qquad a_2 = \text{ReLU}(w_2 a_1 + b_2), \qquad \hat{y} = w_3 a_2 + b_3$$

The loss is mean squared error:

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

You are given the following values:

$$w_1 = 2.0,\ w_2 = -1.0,\ w_3 = 3.0,\ b_1 = 0.0,\ b_2 = 1.0,\ b_3 = 0.0$$

$$x = 1.0, \qquad y = 2.0$$

(a) **(4 points)** Perform the full forward pass. Compute and report the intermediate values a_1, a_2, $\hat{y}$, and $\mathcal{L}$.

Solution:

$$a_1 = \text{ReLU}(2.0 \cdot 1.0 + 0.0) = \text{ReLU}(2.0) = 2.0$$

$$a_2 = \text{ReLU}(-1.0 \cdot 2.0 + 1.0) = \text{ReLU}(-1.0) = 0.0$$

$$\hat{y} = 3.0 \cdot 0.0 + 0.0 = 0.0$$

$$\mathcal{L} = \frac{1}{2}(0.0 - 2.0)^2 = \frac{1}{2}(4.0) = 2.0$$

(b) **(6 points)** Perform the full backward pass using backpropagation. Compute the gradients $\frac{\partial \mathcal{L}}{\partial w_3}$, $\frac{\partial \mathcal{L}}{\partial w_2}$, and $\frac{\partial \mathcal{L}}{\partial w_1}$. Show all intermediate gradient computations clearly.

Solution: Working backwards from the loss:

Gradient of loss w.r.t. $\hat{y}$:

$$\frac{\partial \mathcal{L}}{\partial \hat{y}} = \hat{y} - y = 0.0 - 2.0 = -2.0$$

Gradient w.r.t. w_3:

$$\frac{\partial \mathcal{L}}{\partial w_3} = \frac{\partial \mathcal{L}}{\partial \hat{y}} \cdot a_2 = (-2.0)(0.0) = \mathbf{0.0}$$

Gradient w.r.t. a_2:

$$\frac{\partial \mathcal{L}}{\partial a_2} = \frac{\partial \mathcal{L}}{\partial \hat{y}} \cdot w_3 = (-2.0)(3.0) = -6.0$$

Gradient through ReLU at a_2:
Since the pre-activation input to the second ReLU was $w_2 a_1 + b_2 = -1.0 < 0$, the ReLU is in the *dead* region and its derivative is 0:

$$\frac{\partial \mathcal{L}}{\partial (w_2 a_1 + b_2)} = \frac{\partial \mathcal{L}}{\partial a_2} \cdot \mathbf{1}[w_2 a_1 + b_2 > 0] = (-6.0)(0) = 0.0$$

Gradient w.r.t. w_2:

$$\frac{\partial \mathcal{L}}{\partial w_2} = \frac{\partial \mathcal{L}}{\partial (w_2 a_1 + b_2)} \cdot a_1 = (0.0)(2.0) = \mathbf{0.0}$$

Gradient w.r.t. a_1:

$$\frac{\partial \mathcal{L}}{\partial a_1} = \frac{\partial \mathcal{L}}{\partial (w_2 a_1 + b_2)} \cdot w_2 = (0.0)(-1.0) = 0.0$$

Gradient through ReLU at a_1:
Since the pre-activation input to the first ReLU was $w_1 x + b_1 = 2.0 > 0$, the ReLU is active and its derivative is 1:

$$\frac{\partial \mathcal{L}}{\partial (w_1 x + b_1)} = \frac{\partial \mathcal{L}}{\partial a_1} \cdot \mathbf{1}[w_1 x + b_1 > 0] = (0.0)(1) = 0.0$$

Gradient w.r.t. w_1:

$$\frac{\partial \mathcal{L}}{\partial w_1} = \frac{\partial \mathcal{L}}{\partial (w_1 x + b_1)} \cdot x = (0.0)(1.0) = \mathbf{0.0}$$

(c) **(2 points)** Based on your answer to part (b), what happens to w_1, w_2, and w_3 after one step of gradient descent? What does this imply about the network's ability to learn from this example, and which specific phenomenon is responsible?

Solution: All three gradients are zero, so gradient descent leaves all weights unchanged:

$$w_i \leftarrow w_i - \eta \cdot 0 = w_i \quad \text{for } i = 1, 2, 3$$

The network cannot learn from this example at all. The responsible phenomenon is the **dying ReLU** problem: the pre-activation value at the second layer was negative (-1.0), causing that

ReLU to output zero and block all gradient flow to w_2 and w_1. Even though the first ReLU was active, the zero gradient from the dead second ReLU propagates back and zeroes out all upstream gradients.

(d) **(2 points)** Propose two concrete ways to address the dying ReLU problem identified in part (c), and briefly explain the mechanism behind each.

Solution: Any two of the following for full credit:

Leaky ReLU. Replace ReLU with $f(x) = \max(\alpha x, x)$ for a small $\alpha > 0$ (e.g. 0.01). This allows a small negative gradient to flow even when the unit is not active, preventing the gradient from being completely blocked.

Careful weight initialization. Initialize weights such that pre-activation values are likely to be positive at the start of training (e.g. He initialization). This reduces the chance that units enter the dead region before they have had a chance to learn.

Batch normalization before activations. Normalizing pre-activation values to have zero mean and unit variance makes it less likely that all inputs to a ReLU are consistently negative, keeping more units active throughout training.

Smaller or adaptive learning rates. Large gradient steps can push pre-activation values into the negative region permanently. Using a smaller learning rate or an adaptive optimizer like Adam reduces this risk.

Short Answers: Deep Learning (14 points + 6 extra credit)

10. **(14 points + 6 extra credit.)** You are designing a Convolutional Neural Network (CNN) for digit classification using grayscale images of size 32×32.

(a) **(2 points)** The first convolutional layer uses 8 filters of size 3×3, stride = 1, and no padding. What will be the dimension of the output features?

Solution: Without padding and with stride 1, output size is:

$$\text{Output size} = (32 - 3 + 1) \times (32 - 3 + 1) = 30 \times 30$$

Since there are 8 filters, the full output has shape $30 \times 30 \times 8$.

(b) **(2 points)** After convolution, the output is passed through a ReLU activation. Why is ReLU typically preferred over sigmoid in CNNs?

Solution: ReLU avoids vanishing gradients and is computationally cheaper. It introduces non-linearity without saturating like the sigmoid, helping deeper networks train faster and more effectively.

(c) **(Extra Credit, 3 points)**
Following the above question, we know ReLU is favorable in practice because of many good properties. However, ReLU didn't just come from nowhere! What's the mathematical relationship between ReLU and the sigmoid function?

Hint: the paper mentioned in lecture *Rectified Linear Units Improve Restricted Boltzmann Machines* talked about this!

(d) **(2 points)** A 2×2 max-pooling layer (stride = 2) is applied after the first convolutional layer. What is the new dimensions of the output of the pooling layer?

Solution: Max pooling reduces the spatial dimensions by a factor of 2:

$$\frac{30}{2} \times \frac{30}{2} = 15 \times 15$$

So the new feature map is $15 \times 15 \times 8$.

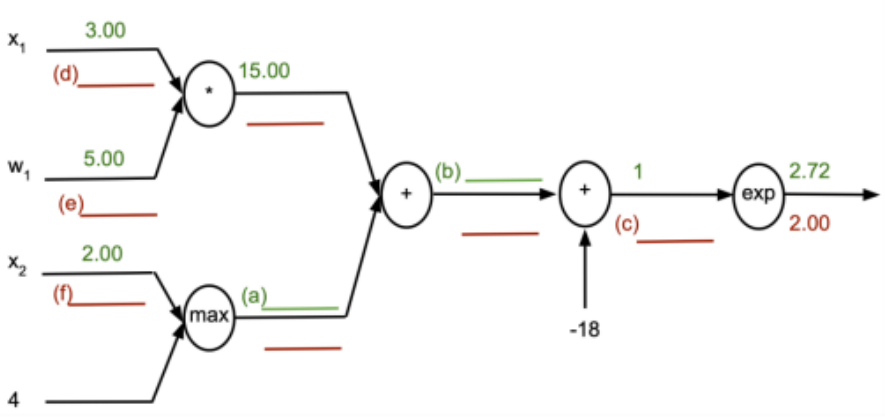
(e) **(4 points)** You are given a single channel 4×4 image and a 2×2 convolution filter with stride 1 and no padding. The convolution has 1 output channel. Let's say we wanted to use an MLP layer instead of a convolution layer. Describe the equivalent weight matrix W for such an MLP layer. As input to the MLP layer, we will flatten the input image like so:

$$\text{Flattened input } \mathbf{x} \in \mathbb{R}^{16} = \begin{bmatrix} x_{(0,0)} \\ x_{(0,1)} \\ x_{(0,2)} \\ x_{(0,3)} \\ x_{(1,0)} \\ \vdots \\ x_{(3,3)} \end{bmatrix}$$

where $x_{(i,j)}$ denotes the pixel at row i and column j in the original 4×4 image. You may use the following convolution matrix if it helps to describe the matrix W

$$\text{Convolution filter} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

(f) **(4 points)** We are represent this neural network using the diagram below:



You are given that the weight of the network currently is $w_1 = 5.00$.

Assume that when the input is $x_1 = 3.00$, $x_2 = 2.00$, the output is $y = 2.72$ and the gradient with respect to the output is 2.00. Using this information, calculate the gradients for the weights of the network w_1.

You are required to fill in the forward pass for (a) and (b) and similarly, you are required to fill in the backward pass gradients for (c), (d), (e) and (f). **Please fill in the the complete gradients after applying chain rule and not just the local gradients**. In case you get anything wrong, we will give you partial credit for the other (unmarked) forward and backward values in the diagram.

(g) **(Extra Credit, 3 points)** Given:

- Ground-truth label $y = 3$ (one-hot vector),
- Softmax prediction for class 3 is $p_3 = 0.1$,
- Loss function is cross-entropy.

Compute the partial derivative of the loss with respect to p_3.

Solution: Cross-entropy loss: $\mathcal{L} = -\log(p_3)$ Gradient: $\frac{\partial \mathcal{L}}{\partial p_3} = -\frac{1}{p_3} = -\frac{1}{0.1} = -10$