Time: 60 minutes Max marks: 20

1. (5 points)

   (a) (1 point) Write the expression for the binary cross-entropy loss and define all the variables.

   (b) (1 point) For classification problems, the target variables $\mathbf{y}$ are usually represented as a *one-hot encoded vector*, say denoted by $\mathbb{1}_i$, with i indicating the index of 1. An alternative to one-hot encoded vectors is obtained by *label-smoothing*, which can be computed as:

$$P(y^{(j)}) = \frac{N-1}{N}\mathbb{1}_i + \frac{1}{N}\mathcal{U}(N).$$

   where $\mathcal{U}(N)$ is a discrete uniform distribution over the N classes. Compute the entropy of $P(y^{(j)})$ in terms of N.

   (c) (1+1+1=3 points) Let an object detector contain a regressor for localizing the object bounding box by the 4-D vector $[x, y, w, h]^\top$. Assume that the Distance-IoU loss is being used to train this object detector. (i) Write the expression for the loss function. (ii) Re-write the loss function in terms of the 4-D output vector of the regressor, i.e., $[x, y, w, h]$. (iii) Can this function be differentiated during backpropagation? If yes, compute the gradient for a non-trivial (i.e., $0 < IoU < 1$) example, else show that it can't be computed.

Solution:

(a) Let $P(y = 1|x) = \widehat{p}$ be the probability of the output variable y being 1, given the input variable x. Assume the target probability is provided to be $p_{gt} = 1$. The binary cross-entropy loss is given by:

$$L_{ce} = -\left(p_{gt}\log(\widehat{p}) + (1 - p_{gt})\log(1 - \widehat{p})\right)$$

(b) $P(y^{(j)})$ will be an N-dimensional vector with the i^{th} entry as $\frac{N-1}{N} + \frac{1}{N^2}$ and all other entries as $\frac{1}{N^2}$. Let $p_i = \frac{N-1}{N} + \frac{1}{N^2}$. The entropy of such a distribution would be:

$$\begin{aligned} H(P) &= -\left(p_i\log(p_i) + \sum_{j=1, j\neq i}^{N}\frac{1}{N^2}\log(\frac{1}{N^2})\right) \\ &= -\left(p_i\log(p_i) - 2\sum_{j=1, j\neq i}^{N}\frac{1}{N^2}\log(N)\right) \end{aligned}$$

(c) (i)

$$L_{DIoU} = 1 - IoU + \frac{d^2}{c^2}$$

Where,

IOU: Intersection over Union of the predicted and ground truth boxes.
d: The distance between the centroids of the predicted and the ground truth boxes.
c: The length of the smallest box containing the two boxes.

(ii) ‘

$$\begin{aligned} x_l &= x - \frac{w}{2}; \\ x_r &= x + \frac{w}{2}; \\ x'_l &= x' - \frac{w'}{2}; \\ x'_r &= x' + \frac{w'}{2}; \end{aligned}$$

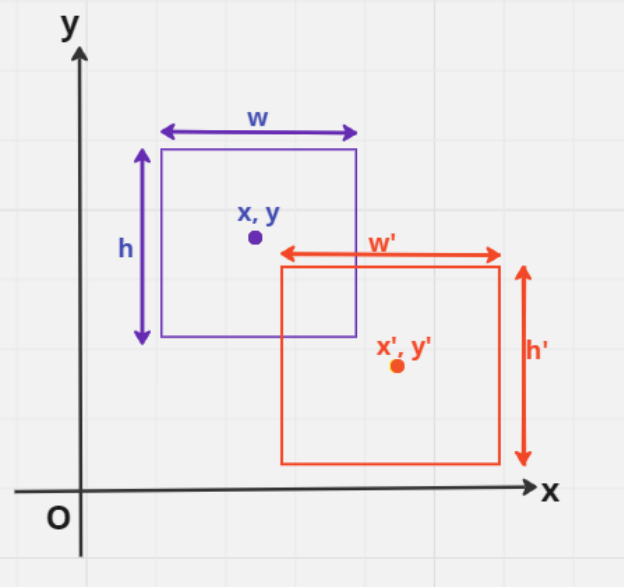

Figure 1: Basic Bounding Box Representation

$$
\begin{aligned}
x_{\min} &= \max(x_l, x_l') \\
x_{\max} &= \min(x_r, x_r') \\
w_{\text{int}} &= \max(x_{\max} - x_{\min}, 0)
\end{aligned}
$$

Similarly,

$$
\begin{aligned}
y_b &= y - \frac{h}{2}; \\
y_t &= y' + \frac{h}{2}; \\
y_b' &= y' - \frac{h'}{2}; \\
y_t' &= y' + \frac{h'}{2}; \\
y_{\min} &= \max(y_b, y_b') \\
y_{\max} &= \min(y_t, y_t') \\
h_{\text{int}} &= \max(y_{\max} - y_{\min}, 0)
\end{aligned}
$$

$$
\begin{aligned}
A_{\text{int}} &= w_{\text{int}} \cdot h_{\text{int}} \\
A_{\text{union}} &= wh + w'h' - w_{\text{int}} h_{\text{int}}
\end{aligned}
$$

$$
\begin{aligned}
d^2 &= (x - x')^2 + (y - y')^2 \\
c^2 &= \left(\max(x_r, x_r') - \min(x_l, x_l')\right)^2 + \left(\max(y_t, y_t') - \min(y_b, y_b')\right)^2
\end{aligned}
$$

(iii) Since there are min-max terms, the gradient can be computed and backpropagated through the computational graph, except at the precise discontinuities.

2. (5 points) Justify your design choices.

   (a) (1 point) You want to increase the receptive field of the features in your convolutional feature maps. Would you prefer a larger convolutional kernel or a pooling / subsampling layer or a larger stride?

   (b) (1 point) You have to design a deep convolutional neural network with many layers. Which activation function would you use: sigmoid or rectified linear unit?

   (c) (1 point) For designing a binary classifier, would you have a sigmoid layer or a softmax layer as the output layer?

   (d) (1 point) A CNN-based semantic segmentation model, like Fully Convolutional Network (FCN) upsamples the last layer's features to generate the semantic segmentation mask. What strategy did FCNs use improve the accuracy (mIoU) using almost the same architecture? What is the possible reason for that strategy to work?

   (e) (1 point) You designed a CNN-based object detector which uses anchor boxes. Why would you wish to use anchor boxes?

   Solution:

   (a) Pooling or larger stride would be preferable to a larger convolutional kernel to maintain a smaller number of learnable parameters. A larger stride would be preferred over pooling to avoid discontinuities (e.g., in max pooling) or fixed averaging while subsampling (in average pooling).

   (b) Rectified Linear Unit (ReLU) to make sure that sufficient gradient magnitudes are maintained to avoid the vanishing gradient problem.

   (c) Sigmoid over softmax for binary classification as it has fewer number of learnable parameters.

   (d) FCNs used a combination of higher-layer features for leveraging semantic information and low-level layers' features to leverage better localization information to improve overall semantic segmentation accuracy in terms of mIoU.

   (e) Anchor boxes serve as initial references for the regressor, which predicts offsets with respect to the anchor box locations[1].

3. (5 points) Justify the statements below and provide one caveat that you would consider while working with this approach.

   (a) Encoder-Decoder architectures are necessary for semantic segmentation.

   (b) The Intersection over Union (IoU) loss is important for CNN-based object detectors.

   (c) CNN-based object detectors need pyramid like architectures to deal with the scale of objects.

   (d) The validation set is important for performance evaluation of machine learning (ML) models.

   (e) The mean Average Precision (mAP) is often used for evaluating the object detection performance by associating predictions with the ground-truth using an IoU threshold of 0.5.

   Solution:

   (a) Encoder-Decoder architectures are helpful for semantic segmentation as the decoder allows gradual upsampling to a higher resolution, thus preserving the finer details of the shape, structure, and boundary of the objects being segmented.

   (b) The IoU loss has been very effective as it captures the overlap, as used at inference time, between the prediction and ground-truth boxes better than L_2 losses that were used with earlier models. However, the IoU loss saturates once the objects have no overlap and, therefore may not allow to learn better features for instances where the predicted and ground-truth boxes do not overlap.

[1] *Note*: These anchor boxes were prevalent in older models (e.g., Faster RCNN, YOLO) that usually used an L_2 type loss for regression. With more modern architectures and losses, studies have shown limited use of anchor boxes when the right kind of loss function (IoU or focal loss).

(c) The scale of objects has traditionally been captured using image pyramids, which can be used as CNNs as well, however, at an increased computation cost. An alternative is to use the different layers of a Deep CNN model. The caveat is that the lower-level features may not have sufficient semantic information, and additional modifications to the learning strategy are necessary to ensure that features at all levels of the pyramid have sufficient semantic information (e.g., in FPNs).

(d) Validation set is the surrogate used for the test set, which measures the model's performance on *unseen* data. The validation set should be similar to the data we expect at test time, or else our model's performance estimates would be off.

(e) The mAP is good for quantifying the object detection performance as it helps reporting true detections, false alarms, and misses in one number. However, an association based on only a single IoU threshold of 0.5 is limiting as it treats excellent predictions (with $IoU \simeq 1$) and average predictions as the same. Secondly, mAP deteriorates more severely for smaller objects than larger objects due to its reliance on IoU.

4. (5 points) A 2-D point $[2,3]^\top$ is first rotated by an angle of $\frac{\pi}{3}$ and then translated by $[1,1]^\top$. (i) What are the new coordinates? (ii) Is there a transformation that would map $(2,3)$ to the same point as in (i), but by translating first and then rotating? If yes, find the translation vector $\mathbf{t}$ and the corresponding rotation matrix $\mathbf{R}$. If not, then justify why it is not possible. {*Hint*: You may keep calculations to fractions / approximate.}

Solution:

The transformation of rotation followed by translation would be:

$$\mathbf{T} = \mathbf{T_t R}$$

$$\mathbf{T} = \begin{bmatrix} \cos\pi/3 & -\sin\pi/3 & 1 \\ \sin\pi/3 & \cos\pi/3 & 1 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\pi/3 & -\sin\pi/3 & 0 \\ \sin\pi/3 & \cos\pi/3 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The transformed point is:

$$\begin{bmatrix} 2 - \frac{3\sqrt{3}}{2} \\ \sqrt{3} + \frac{5}{2} \\ 1 \end{bmatrix} = \begin{bmatrix} 1 - \frac{3\sqrt{3}}{2} + 1 \\ \sqrt{3} + \frac{3}{2} + 1 \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\pi/3 & -\sin\pi/3 & 1 \\ \sin\pi/3 & \cos\pi/3 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 3 \\ 1 \end{bmatrix}$$

(ii) We want $\mathbf{T} = \mathbf{R'T_{t'}}$. Note that $\mathbf{T}$ needs to remain the same as our target point has not changed. The rotation also remains the same (*why?*), i.e., $\mathbf{R'} = \mathbf{R}$, therefore:

$$\mathbf{T} = \begin{bmatrix} \cos\pi/3 & -\sin\pi/3 & 1 \\ \sin\pi/3 & \cos\pi/3 & 1 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos\pi/3 & -\sin\pi/3 & 0 \\ \sin\pi/3 & \cos\pi/3 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & t'_x \\ 0 & 1 & t'_y \\ 0 & 0 & 1 \end{bmatrix}$$

We have

$$t'_x \times \cos\pi/3 - t'_y \times \sin\pi/3 = 1$$
$$t'_x \times \sin\pi/3 + t'_y \times \cos\pi/3 = 1$$

Solving these two equations for t'_x and t'_y, you get $(t'_x, t'_y) = (\frac{1+\sqrt{3}}{2}, \frac{1-\sqrt{3}}{2})$. Plugging these values in will show that the same transformation matrix can be achieved, which will transform the point $(2,3)$ to the same point as in (i).

5. (*Extra credit*: 3 points) Show that the Kullback-Leibler (KL) Divergence for PMFs is not symmetric and can take infinite values. The Jensen-Shannon Divergence (JSD) is given by $JSD(P||Q) = \frac{1}{2}KL(P||M) +$

$\frac{1}{2}KL(Q||M)$, where $M = \frac{P+Q}{2}$ is a mixture distribution of the two. Show that, for PMFs, JSD is symmetric and will not take infinite values.

Solution:

The KL Divergence is not symmetric because it is defined as $KL(P||Q) = \sum_x (P(x)\log(P(x) - Q(x))$. It is obvious that swapping P and Q will lead to different values of the function. The KL Divergence can take infinite values whenever $Q(x) = 0$ and $P(x) \neq 0$.

$$JSD(P||Q) = \frac{1}{2}KL(P||M) + \frac{1}{2}KL(Q||M) \tag{1}$$

$$JSD(Q||P) = \frac{1}{2}KL(Q||M) + \frac{1}{2}KL(P||M) \tag{2}$$

From (1) and (2), we can show:

$$JSD(P||Q) = JSD(Q||P) - (3)$$

Also, on expanding $JSD(P\|Q)$:

$$\begin{aligned}
JSD(P\|Q) &= \frac{1}{2}KL\left(P\middle\|\frac{P+Q}{2}\right) + \frac{1}{2}KL\left(Q\middle\|\frac{P+Q}{2}\right) \\
&= \frac{1}{2}\sum_x P(x)\left[\log P(x) - \log\left(\frac{P(x)+Q(x)}{2}\right)\right] + \frac{1}{2}\sum_x Q(x)\left[\log Q(x) - \log\left(\frac{P(x)+Q(x)}{2}\right)\right] \\
&= \frac{1}{2}\sum_x \left\{P(x)\log P(x) - P(x)\log\left(\frac{P(x)+Q(x)}{2}\right) + Q(x)\log Q(x) - Q(x)\log\left(\frac{P(x)+Q(x)}{2}\right)\right\} \\
&= \frac{1}{2}\sum_x \left\{P(x)\log P(x) + Q(x)\log Q(x) - [P(x)+Q(x)]\log\left(\frac{P(x)+Q(x)}{2}\right)\right\} \\
&= -\frac{1}{2}\left(H(P) + H(Q)\right) - \sum_x \frac{P(x)+Q(x)}{2}\log\left(\frac{P(x)+Q(x)}{2}\right) - (4)
\end{aligned}$$

Here, H(P) and H(Q) are the entropies of the PMFs, P(x) and Q(x), respectively.
Hence,

From 3, We can see that JSD is symmetric.

From 4, We can see that JSD can not take infinite values as it can be expressed in terms of the entropy of P, Q, and $\frac{P+Q}{2}$, which are all finite quantities.