Assignment 5

November 2021

Instructions

- This assignment should be completed individually.
- Do not look at solutions to this assignment or related ones on the Internet.
- The files related to the assignment are present in `lab5-rollno.zip` folder. Extract it and upload it on moodle in the same .zip format after completion and after replacing the string "rollno" with your actual roll number. For example, if you roll number is 00405036, then single zip folder that you will upload will be named "lab5-00405036.zip". Also collate all the CS337 based theory solutions into ONE pdf file named `answers.pdf`. Include `answers.pdf` inside the zip folder mentioned above and submit the zip folder.
- Answers to all subjective questions need to be placed in single pdf `answers.pdf` including all plots and figures and uploaded.
- Only add/modify code between `TODO` and `END TODO` unless specified otherwise. You must not import any additional libraries.
- Files to submit - `layers.py`, `nn.py`, `trainer.py` and `model.p`
- This Assignment carries a total of **10** marks for CS335 Lab and **22** marks for CS337 theory.

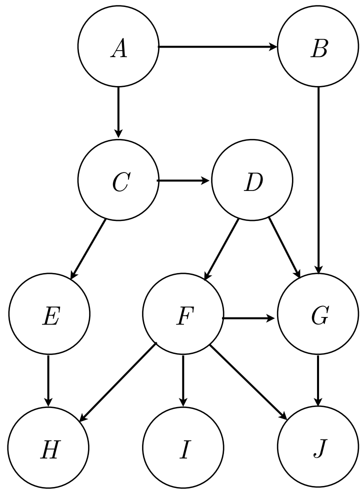
1 CS 337: Directed Graphical models

1.1 D-Separation

For the Directed graphical model given in the Figure 1, answer the conditional independence queries. If the answer is `No`, mention atleast one path that is not blocked. If the answer is `Yes`, mention why all paths are blocked having observed the conditioning variables. ($10 \times 0.5 = 5$ marks)

1.2 Probability distribution

Write the expression for the joint probability $P(A, B, C, D, E, F, G, H, I, J)$. Your expression should contain 10 conditional probability terms such that the joint probability factors over the 10 terms. (2 marks)



(a) $C \perp B$?

(b) $C \perp B \mid A$?

(c) $C \perp B \mid A,J$?

(d) $C \perp B \mid A,J,D$?

(e) $C \perp G$?

(f) $C \perp G \mid B$?

(g) $C \perp G \mid B,D$?

(h) $C \perp G \mid B,D,H$?

(i) $C \perp G \mid B,D,H,E$?

(j) $B \perp I \mid J$?

Figure 1: D-Separation Questions

1.3 Number of parameters required to learn the probability distribution

Assume that all the nodes in the DGM represent Bernoulli Random variables. i.e. each node takes value 1 with probability θ and value 0 with probability $1-\theta$. Further assume that we know the node I always takes what F takes (i.e., I = F) and the node H takes the value E logical OR F (*i.e.*, $H = E \vee F$). We know that learning Bernoulli random variable amounts to learning just one parameter θ. i.e. $P(A = a) = \theta^a(1-\theta)^{1-a}$ Specify the minimum number of parameters we need to learn the joint probability distribution. In process of doing so, complete the following table:

Prob. distribution	# Parameters
P(A)	1
$P(A,B,C,D,E,F,G,H,I,J)$	Total parameters

(3 marks)

2 CS337: CNN Theory Questions

In this problem, assume that every input image has the same size: 1024×1024**.**

Task 1: Detecting Image Containing
Single Object

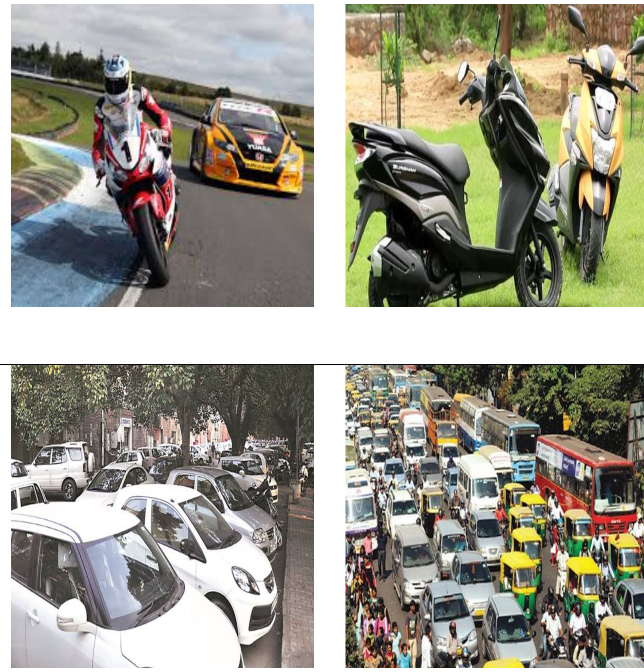
Figure 2: List of 1024×1024 images for Task1.

1. **Task 1:** In the class we discussed the Convolutional and pooling layers of a CNN and motivated the kernel (sparse interaction, patches, strides) for classification of images. Suppose our input image contains exactly one vehicle (either a bicycle or a car or a motorbike or any one of N vehicles) and we have trained a CNN-based network to distinguish between images based on the kind of vehicle that the image contains. Specifically, the output layer of our network consists of a soft-max layer that can classify an image into one of N vehicular categories. Thus, our CNN-based network is trained to distinguish that the image in the first row, first column of Figure 2 is a car and that the image in the first row, second column of Figure 2 is a motor-bike.

   Roughly depict or describe for each of the images of the 3 columns of Figure 3 what the different components/operators of our trained CNN-based network would extract in the intermediate layers to help distinguish between images containing different vehicles, and how the properties of CNNs would help handle differences in the objects' location and size.[1] Here all the images are actually reproduced from the first row of Figure 2. (5 marks)

Figure 3: Explain roughly how the CNN-based network helps address/correctly classify the images here. Recall that each image is of the same size, *viz.*, 1024×1024

[1] that is, give correct results even when the location or size of an object in the image changes



Task 2: Detecting Separated Objects

Task 3: Detecting Overlapping Objects

Figure 4: List of 1024×1024 images for the three tasks in this problem.

2. **Task 2:** Suppose we now have images that contain a combination of multiple vehicles but well separated from each other within the same image, as illustrated in the images in the second row of Figure 4. How would you modify your CNN-based network from Task 1 to detect different objects in the same image? Identify any important limitation in your solution. (4 marks)

3. **Task 3:** Now suppose we need to detect overlapping vehicles within the same image. That is, we now have images that contain a combination of multiple vehicles that are NOT well separated from each other within the same image, as illustrated in the images in the third row of Figure 4. This kind of phenomenon is called occlusion. Suggest how you will modify the CNN-based network to handle Task 3. Identify any important limitation in your solution. (3 marks)

3 CS 337: Feed Forward Networks

In this problem, we will implement a feed forward network to predict digit from a image (using MNIST dataset) and flower type from flower dataset given in the `data` folder. In MNIST dataset each element in 28x28 2D array and in flowers dataset each element is a vector of length 2048.

In `assignment_5.ipynb`, complete the functions `fit` to train a feed forward network and `predict` to load the trained weights and return labels for the given input. (10 marks)

(i) To implement a feed forward networ k complete `forward` and `backward` functions of `FCLayer`, `ActivationLayer` and `SoftmaxLayer` classes.

(ii) Complete functions `sigmoid`,`sigmoid_prime`,`tanh`,`tahn_prime`, `relu` and `relu_prime` functions to implement respective activation functions and their derivatives and invoke them in `forward` and `backward` functions of `ActivationLayer`.

(iii) Complete functions `mse`,`mse_prime`,`cross_entropy` and `cross_entropy_prime` functions to implement respective loss functions and their derivatives and invoke them while training the feed forward network.

There are several other helper functions such as `FlattenLayer`, `preprocessing` etc. which could be utilised to transform the inputs to feed forward network in an appropriate manner. You may define other helper functions if you need.