CS 335
Assignment 5

Aakriti

Due 20th November 2021

Contents

1 CS 337: Directed Graphical models

1.1 D-Separation

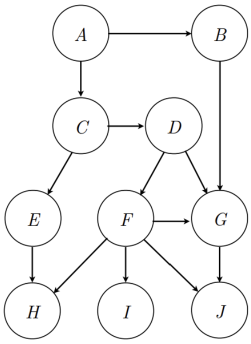


Figure 1: D-Separation Questions

No.	Conditional Independence Query	Answer	Path Not Blocked
(a)	$C \perp B$	No	$C \longleftarrow A \longrightarrow B$
(b)	$C \perp B \mid A$	Yes	The paths connecting C to B, must either pass through A (which is blocked due to A) or from H, G or J as a Head-to-Head node and thus, all 3 of them are blocked.
(c)	$C \perp B \mid A, J$	No	$C \longrightarrow D \longrightarrow G \longrightarrow J \longleftarrow G \longleftarrow B$ (since J is given but is Head-to-Head node).
(d)	$C \perp B \mid A, J, D$	Yes	The only possible path mentioned above now gets blocked since D is given to us.
(e)	$C \perp G$	No	$C \longrightarrow D \longrightarrow G$
(f)	$C \perp G \mid B$	No	$C \longrightarrow D \longrightarrow G$
(g)	$C \perp G \mid B, D$	Yes	The paths connecting C to G, must either pass through B or D (both are blocked), or from H as a Head-to-Head node (and therefore, blocked).
(h)	$C \perp G \mid B, D, H$	No	$C \longrightarrow E \longrightarrow H \longleftarrow F \longrightarrow G$
(i)	$C \perp G \mid B, D, H, E$	Yes	The only possible path now gets blocked as it passes through E.
(j)	$B \perp I \mid J$	No	$B \longleftarrow A \longrightarrow C \longrightarrow D \longrightarrow F \longrightarrow I$

1.2 Probability distribution

We know that,

$$P(x_1, x_2, \ldots, x_N) = \prod_{i=1}^{N} Pr(x_i|\texttt{parent}[x_i])$$

$$P(A,B,C,D,E,F,G,H,I,J) = P(A) \times P(B|A) \times P(C|A) \times P(D|C) \times P(E|C) \times P(F|D) \times P(G|D,B,F) \times P(H|E,F) \times P(I|F) \times P(J|F,G)$$

1.3 Number of parameters required to learn the probability distribution

Term No.	**Prob. distribution**	**# Parameters**
1	P(A)	1
2	P(B|A)	2
3	P(C|A)	2
4	P(D|C)	2
5	P(E|C)	2
6	P(F|D)	2
7	P(G|D,B,F)	8
8	P(H|E,F)	0
9	P(I|F)	0
10	P(J|F,G)	4
	P(A, B, C, D, E, F, G, H, I, J)	23

For each configuration of the parents, node is a Bernoulli random variable. For example, for P(B|A), for every value of parent A, B will have unique parameters. So, a node with N parents will have 2^N parameters. So, P(A) has 1 parameter; each of nodes B, C, D, E, F have 1 parent and so $2^1 = 2$ parameters; node G has 3 parents and thus, 2^3 parameters and J has 2 parents so has $2^2 = 4$ parameters. We are given that 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). So, nodes H and I have no parameters.

2 CS337: CNN Theory Questions

2.1 Task 1

Task 1: Detecting Image Containing Single Object

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

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

- A CNN works by detecting different features present in the image. The initial layers detect simple features like edges and simple curves, and then the deeper layers can detect the combination of these simple features.
- The output of the convolution layers are activation maps which get activated on detecting the presence of a feature.
- Thus, for example, for the images given in Figure 2, the initial layers can detect the wheels (present in both car and bike). However there would be differences in that too.
- Also for cars, the number plate (bonnet), doors of car, two headlights in front, the window shield would be detected by the CNN as features and since the combination of all these features are present, the first and third image would be classified as cars.
- On the other hand, if features like two wheels, bike handle, seats of bike, petrol tanks are detected, the CNN would classify the object as a bike since it has been trained to learn such features and classify objects depending on the combination of features present.
- The sparse interactions help in reducing the number of parameters, since to detect a feature we only need the local pixel values and not the whole image.
- CNN is robust to positions of objects due to sharing of parameters (same for each patch of input image), hence CNN can detect the object if it is present in any part of the image.

- Also the Max Pooling layers causes translation invariance and takes care of different sizes of objects as well by causing magnification of signals. Hence if a feature is present anywhere in the image, the activation map for that gets activated and hence CNN is able to detect it.
- Hence translation equivariance and invariance helps classify images even if objects are present in different positions in images with different sizes.

Task 2: Detecting Separated Objects

2.2 Task 2

- We can start by building classifiers for each object , i.e. a classifier for each vehicle , the output of each classifier would be between 0 to 1 denoting the probability of the object being present in the image, i.e. instead of using softmax, we use sigmoid for each object.
- Hence the CNN can combine different features and if it detects the presence of a specific set of features (corresponding to say car), then the output corresponding to car is high. If it detects features of the bike as well, output corresponding to the bike is also high.
- For training such CNN , we can use training data labelled with all the objects present in it.
- A limitation of this method would be that the number of network parameters would be quite high and also different features can interfere together , i.e. suppose some feature of a car and a cycle can combine and give us the illusion that a three wheeler vehicle like an autorickshaw is present (just an example). Hence this can lead to incorrect results.
- To do this more efficiently , we can divide the image (1024*1024) into small grids, say 128*128 , then there would be 64 such grids. For each grid we can detect whether an object is present or not, if present then what its centre is and the bounding box for the object , height and width of the bounding box.
- We assume that annotated training data is available to us and CNN is trained using this . Penalty is imposed on wrong classification of objects and cost is associated with incorrect classification of bounding box dimension as well.
- In the end, we can detect the presence of different objects as well as their location using this approach. Since the objects are sufficiently spaced, we can keep large grid boxes hoping that no 2 objects would coincide in a single grid box.
- An important limitation of this approach is that 2 objects can fall in the same box (if grid size is large) or an object may not be enclosed in the grid box completely (small grid size). Hence there is a trade-off between increasing grid size and decreasing collision chances.

- Also, even if we stick with a fixed grid size, we can combine the bounding boxes of different grids to form a large bounding box for the whole image (using similar features obtained in these grids) , this can help when one object spans multiple grid boxes, since we are given that different objects are well separated.

Task 3: Detecting Overlapping Objects

2.3 Task 3

- We improvise upon the previous approach of bounding boxes and dividing image into grids.The issue with the fixed grid size approach would be, that we can't simply combine grid boxes to form larger bounding box since now objects can be overlapping and hence we won't be able to distinguish while combining.
- Also the centre of two objects may fall into the same grid cell.
- Hence here we employ different box sizes and shapes for different vehicles , i.e. for trucks we keep large box sizes (square), for bikes we keep rectangles (horizontal and vertical), hence different aspect ratios for different objects. This approach is used in algorithms like YOLO.
- Hence even if a bike and truck are overlapping and their centre is close, they'll get classified to the grid cell with the corresponding aspect ratios and won't interfere in our detection.
- However one object can get assigned to different bounding boxes (or grid cells) here as well, which can be resolved by assigning the box with max probability to the object.
- Limitation of this approach is that if two vehicles with the same aspect ratio overlap, then they'll fall in the same cell with the same aspect ratio and hence we won't be able to detect which object is present in that cell.
- In all these tasks we assume that annotated training data is available (consistent with the model).

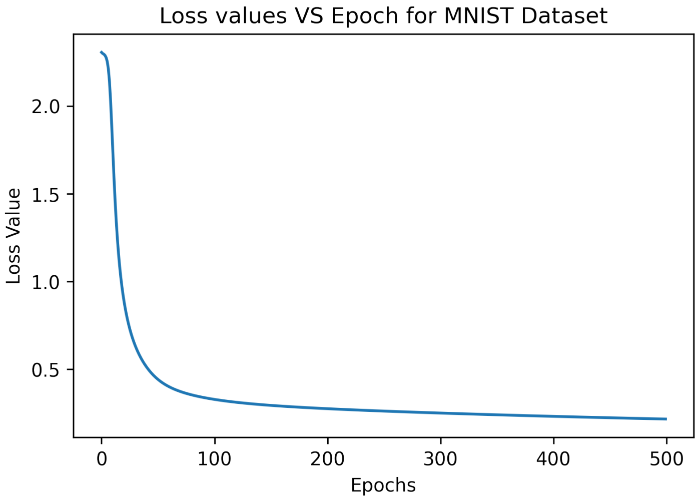
3 CS 337: Feed Forward Networks

Refer to `Assignment_5.ipynb` for code of the functions implemented to train a feed forward network and predict to load the trained weights and return labels for the given input.

Hyperparameters -

- One hidden layer with 48 nodes and ReLU activation function
- Number of epochs = 500
- Learning rate = 0.02
- Batch size = 16

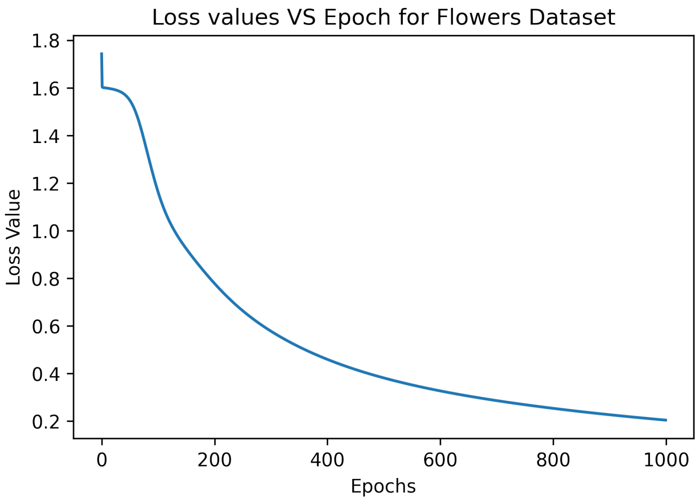
Here are the results of the code -




```
MNIST Dataset - Training Results

Precision: 0.979759281644743
Recall: 0.9797957828358836
F1-score: 0.9797715005526317
Accuracy: 0.9799333333333333
```

Figure 4: Results on MNIST Dataset




```
Flowers Dataset - Training Results

Precision: 0.9906494735567464
Recall: 0.9903011633291362
F1-score: 0.9904697849435445
Accuracy: 0.9799333333333333
```

Figure 5: Results on Flowers Dataset