Assignment 5

Devansh Jain, 190100044

20th Nov 2021

Contents

1 CS 337: Directed Graphical models

1.1 D-Separation

(a) $C \perp B$?
No - CAB is not blocked.

(b) $C \perp B|A$?
Yes - CAB is now blocked and all other paths have a Head-Head node which isn't A.

(c) $C \perp B|A, J$?
No - $CDGB$ is now not blocked as descendent of Head-Head node G is J.

(d) $C \perp B|A, J, D$?
Yes - $CDGB$ is now blocked because of D.

(e) $C \perp G$?
No - $CABG$ and CDG are not blocked.

(f) $C \perp G|B$?
No - CDG is not blocked.

(g) $C \perp G|B, D$?
Yes - $CABG$ and CDG are now blocked and all other paths have a Head-Head node.

(h) $C \perp G|B, D, H$?
No - $CEHFG$ is now not blocked as Head-Head node is H.

(i) $C \perp G|B, D, H, E$?
Yes - $CEHFG$ is now blocked because of E.

(j) $B \perp I|J$?
No - $BGFI$ is not blocked as descendent of Head-Head node G is J.

1.2 Probability distribution

$$\mathbb{P}(A, \cdots, J) = \prod_X \mathbb{P}(X|\text{parents of } X)$$

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

1.3 Number of parameters required to learn the probability distribution

Except for H, I, all conditional probability terms would require $2^{\#\text{Parents}}$ parameters.

Conditional probability term for I would require zero parameter as $P(I = 1 | F = 1) = 1$ and $P(I = 1 | F = 0) = 0$. Similarly, conditional probability term for H would require zero parameter.

Prob. distribution	# Parameters
$\mathbb{P}(A)$	1
$\mathbb{P}(B|A)$	2
$\mathbb{P}(C|A)$	2
$\mathbb{P}(D|A)$	2
$\mathbb{P}(E|C)$	2
$\mathbb{P}(F|D)$	2
$\mathbb{P}(G|B, D, F)$	8
$\mathbb{P}(H|E, F)$	0
$\mathbb{P}(I|F)$	0
$\mathbb{P}(J|F, G)$	4
$\mathbb{P}(A, B, C, D, E, F, G, H, I, J)$	23

2 CS337: CNN Theory Questions

2.1 Task 1

Usually a CNN based network has multiple small kernels (usually 3 or 5 in size) at each level which extract features.

In Fully connected Neural network, the first layer is linear with respect to the input and the complexity increases as we go forward.

Similarly in the CNN based network, the first layer is able to extract simpler features like edges. As we go forward, the network is able to extract features like curves, shapes.

Once these features are extracted, the following layers are fully connected layers (with dropout) to capture relation between these features.

This finally ends with a level with N nodes after which we apply a soft-max layer to get the category.

Use of Kernels provide sparse interactions, so the features a spatially localized, i.e., we can detect shapes irrespective of what is present elsewhere.

"Convolution is not naturally equivariant to some other transformations, such as changes in the scale or rotation of an image. Other mechanisms are necessary for handling these kinds of transformations."[1]

Pooling is the most important component in terms of making the network invariant to location and size of object.

"The use of pooling can be viewed as adding an infinitely strong prior that the function the layer learns must be invariant to small translations."[1]

This invariance carries through multiple layers makes the network handle differences in object's location.

"Because pooling summarizes the responses over a whole neighborhood"[1]

Because of this summarization, the network can handle different sizes of object and if trained properly even rotation in object.

It also depends on the training method. Without sufficient data, the network won't be able to learn such invariances and may over-fit to think that the object is located mostly in one part of image.

To prevent this, we perform data augmentation, i.e. we provide same images with the small translation, scaling and rotation applied to it.

For the given three pictures, the network would extract features like curves and shapes. The pooling layers and the fully connected layer would evaluate the first and last image (car images) as same as they would detect features which are similar to what it has learnt to classify as car. For the bike image, it will evaluate as bike as it's features match those of training set of bike and is different from other vehicles like car. The soft-max layer at the end would output the most confident object detected which would be car, bike, car respectively.

---

[1] Deep Learning. Yoshua Bengio, Ian J. Goodfellow, Aaron Courville

2.2 Task 2

By "detect different object", I am simplifying the task and interpreting it as given that the image has K out of N objects, we are asked to identify the K objects.

This can be handled by changing the last layer from soft-max layer (returns one-hot encoding of size N) to layer which returns a vector of size N where each object detected is marked with 1 and rest 0.

We can retrain the model, the objects which are detected would have high values in second last layer and which can then be converted to the required vector.

The limitation of the above network is that some objects might dominate and may cause noisy detection in some other object classifiers and may give inaccurate results. Like, a truck is likely to give some noisy value to the bus classifier as well. So an image with two trucks and one bike may cause the noise in bus classifier to make it seem like truck and bus are present and not truck and bike.

If we are asked to detect objects and not given number of objects present in the image, then we can use Region-based CNN (RCNN). As the images aren't overlapping, we don't have to worry about a region giving multiple objects.

The major challenge of this approach is efficiently defining regions of interest (RoIs). This approach is slow for training as well as inference - faster RCNN models have been proposed to tackle this issue.

2.3 Task 3

One simple approach is simply extend the model in Task 2 and train the network to classify partial images of an object correctly. RCNN would be better here as each object would take small portion of the image and so finding RoIs would be easier with heat-maps. This however would reduce models accuracy a lot and would require a lot of data to prevent over-fitting.

An approach proposed in an IEEE 2017 paper[1] showed training with proper dataset and applying small spatial support using Regularization helps in detecting objects with occlusions.

A recent paper submitted to arXiv[2] proposed using vMF kernel on the feature maps from CNN based network to generate occlusion likelihood and thus detect occlusions followed by classification ignoring these occluded features.

---

[1]Increasing CNN Robustness to Occlusions by Reducing Filter Support, Elad Osherov and Michael Lindenbaum, `https://ieeexplore.ieee.org/document/8237329`

[2]Compositional Convolutional Neural Networks: A Deep Architecture with Innate Robustness to Partial Occlusion, Adam Kortylewski, Ju He, Qing Liu, Alan Yuille, `https://arxiv.org/abs/2003.04490`