DL End Sem

Sayam Kumar
S20180010158

Set-C

Q-1 ⓐ Explain YOLO with the help of diagram

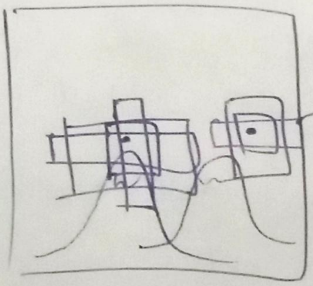
YOLO is a single stage object detection solution. It goes directly from a single input image to these scores (mainly bounding boxes)

→ for each grid cell, it produces B bounding boxes and their confidence scores. The confidence score is given IoU (Intersection over Union)

One bounding box = $\begin{bmatrix} x \\ y \\ w \\ h \\ \text{confidence} \end{bmatrix}$

x, y = center of box

w, h = width and height



Architecture of YOLO → we use 3x3 filters along with 1x1 to capture spatial trends. It is very efficient with respect to training.

Sayam Kumar
S20180010158

(4) YOLO different for Faster-RCNN

(i) Each cell grid in YOLO predicts two boxes and can only have one class. This prevents multi class object detection compared to faster RCNN

(ii) In YOLO, localization accuracy is small due to coarse features and even on smaller boxes

iii) Since YOLO is single shot object detection, it has 7 times speedup than faster RCNN during training and 148 times speedup during testing time

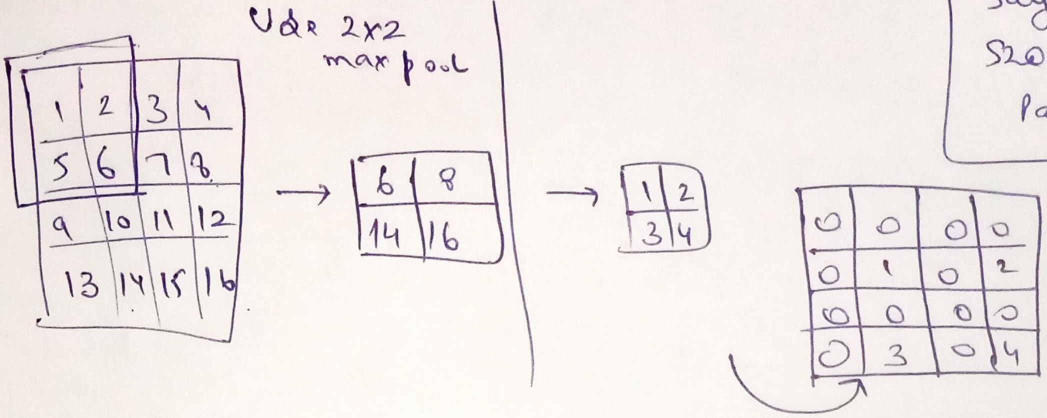
(c) Explain Max-Unpooling and how it is useful for image segmentation.

Ans In max pooling, we ~~do~~ take a max of all input points covered by the kernel size.

→ In max unpooling, we remember the location of max element to segment the image we expect the model to learn

Example next page

Sayam Kumar
S20180010158
Page 3



max unpooling

max unpooling works an unsampler.
In image segmentation, this helps in finding regions of max importance that are closer nearby due to sparse connectivity introduced by kernels.

---

d) Explain how constrastive learning helps self supervised learning?

Ans → Constrastive methods aim at encouraging representations of transformed versions of different objects.

Eg → I am making an end to end DL system for image classification for facials.

I need to make sure that network learns the difference between twins and penalises in the direction of making ~~+~~ similar object dissimilar and disimilar object similar.

eg, Siamese network.

Also we can make use of softmax activation in constrastive learning.

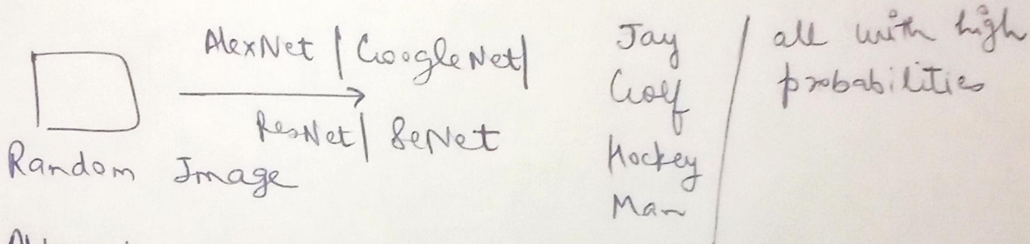
Sayam kumar
S20180010158

Q.2 (a) Example universal adversial perturbations with diagrams.

Ans The main idea of universal adversial perturbations is

(1) finding an image independant ~~perturbati~~ vector that causes all images to misclassified at a high probability.

(2) The idea is if we have an image that satisfies the above criteria, then many deep CNN architectures can be fooled by same image



Approach → i) Start with $r = 0$

ii) Cycle through training examples

→ if $x_i + r$ is misclassified skip to x_{i+1}

iii) Find min Δr that takes $x_i + r + \Delta r$ to another class.

and generate an image

Sayam Kumar
S20180010158
Page 5

(5) Explain Cycle GAN with diagram and objective function. Role of different networks

Ans For image to image translation,

let's say we are moving to coloured images from grayscale images. It is for style transfer.

Given two domains X (input) and Y (output)

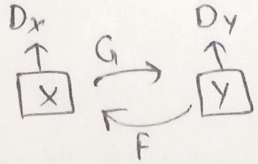
→ Train two generators F and G and two discriminators D_X and D_Y.

→ G translates from $X \to Y$ and F from $Y \to X$

→ D_X recognize image of X and D_Y recognize images of Y.

Consistency

We want $F(G(x)) = x$ and $G(F(y)) = y$



Loss

$$L_{gan}(D_Y) = E_{y \sim data(Y)}\left[(D_Y(y) - 1)^2\right] + E_{x \sim Pdata(x)}\left[D_Y(G(x))\right]^2$$

Discriminator $$L_{gan}(D_X) = E_{x \sim data(x)}\left[(D_X(x) - 1)^2\right] + E_{Y \sim Pdata(Y)}\left[D_X F(y)\right]^2$$

Generator $$E_{x \sim Pdata(x)}\left[D_Y G(x) - 1\right]^2 + E_{Y \sim Pdata(Y)}\left[D_X (F(y) - 1)^2\right]$$

+ same for Y.

Generator networks are encoder decoder networks.

and discriminator are based on Patch GANs

Sayam Kumar
S20180010158

(c) Limitations of Cycle GAN model

(i) Cannot clearly distinguish b/w very similar objects. Eg→ apple/orange

ii) cannot handle shape changes

iii) Cannot handled image outside of training set. It doesnot generalize well.

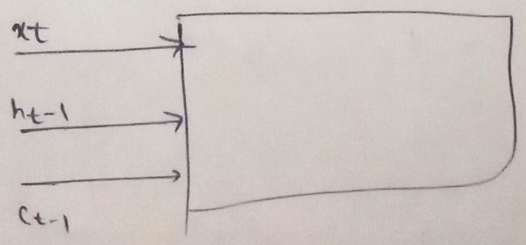
(iv) Doesnot order any transformation applied on objects during pre-processing step.

(v) Take long time to train. Two generators and two discriminators.

Q-3 Sequence Learning

(a) LSTM with diagram. Metion purpose of different gates.

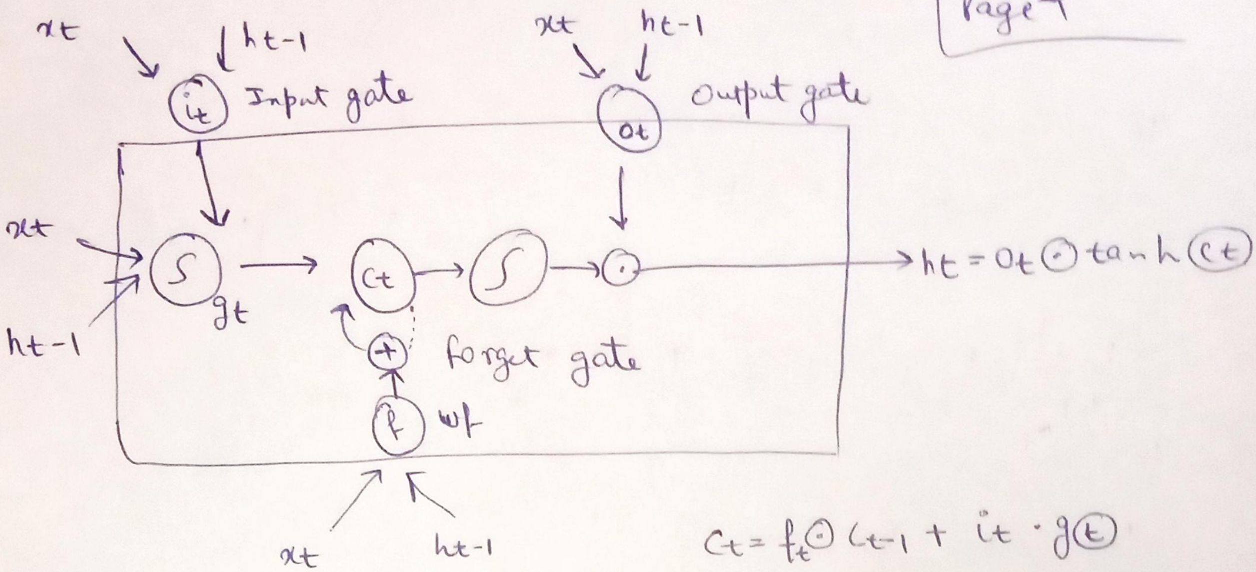
Answer- The main drawback of RNN is their gradient vanishing problem because it is unable to capture long term dependencies. LSTM adds a memory cell that is not subject to matrix multiplication. And this avoids gradient decay



All gates are explained on next page →

LSTM gates

Sayam Kumar
S20180010158
Page 7



$C_t = f_t \odot C_{t-1} + i_t \cdot g(t)$

<u>forget gate</u>

$$C_t = \underbrace{f_t \odot C_{t-1}}_{\text{previous}} + \underbrace{i_t \odot g_t}_{\text{current}}$$

x_t = Input

h_{t-1} = Output of Last LSTM

C_{t-1} = memory unit

<u>Output gate</u> ⇒

$$h_t = O_t \cdot \tanh(C_t)$$

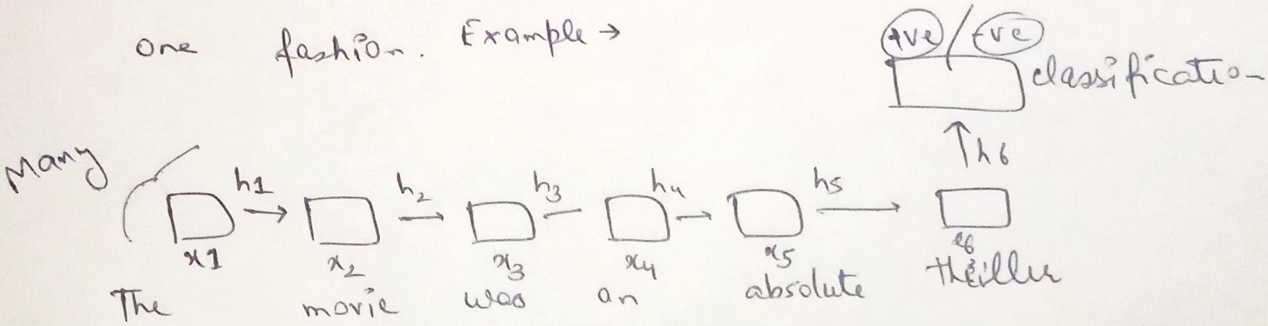
(Apply tanh activation)

Multiple LSTM cells can be rolled out to capture long term dependencies via Ct. While xt serves as an input and ht serves as an output from previous LSTM block. It increases computation with respect to RNN Nothing but yields better performance.

Q-3 b(2) How RNN based models can be used for sentiment classification with diagram and examples →

Sayam Kumar
S20180010158
Page 8

Ans The idea is use RNN in a many to one fashion. Example →



The idea has been used for Twitter sentiment classification challenge.

Concept RNN are great in capturing essence of a sentence provided the entire sequence is judged progressively. So, that's why we use multiple RNN blocks rolled out to capture the sentiment during training. We can consider a max sentence length to avoid analysing large sentences.

Example outputs

1. The cricket match ~~is~~ was good +ve
2. The floods destroyed many villages it went through -ve

Bonus we can use pre trained word embeddings to improve RNN performance.

Sayam Kumar
S20180010158
Page 9

Q-4 (a) (D) Swish is an activation function

(ii) (C) Goog GoogleNet uses inception

(iii) (B) 28x28x96

(iv) (A) with kernel trick as dual optimization problem

(v) (D) a very high learning rate leads to faster convergence of model