21st sep

DL
- ANN Model
- CNN " 3 image
- RNN " } Sequential data
- Transfor " } Sequential data
- GNN " for seq → Graph data

Computer Vision

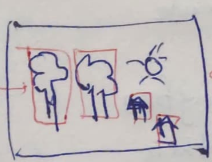
Before Start (Q) How to create model around Images
→ Input data is Image.
→ (Q) What are the use case we solve by CV ?

(Q) What is Exactly Image data contain.

↳ ① Image classification → only 1 type of image are interest

T C ✓ H
T C ✓ H

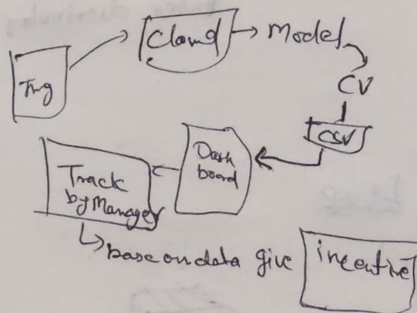
② Object detection & Localization



Means आप उनको Boundry दे रहे हो तो Localization

← also सारे Image में क्या है सब Detect करना
Multiple objects
2 Tree,
2 House
1 Sun

eg: Traffic
: CCTV
: Warehouse
: On Self-Availability →



↳ base on data give incentive

③ Image Segmentation

④ Siamese networks
↳ comparision b/w 2 images

img1, img2 → Same or Not ✓ ✗

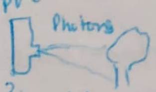
⑤ GAN (Generative Adversal Networks
↳ create Synthetic images

→ Control Level
→ DSLR effect → in Phone
↳ User Control
↓
in pixel level work

all Solve by CNN Model

(Q) What is image ?
↳ 3D
0-255
RGB (

PVC



3 types of Photo Valtic cell
1 - Red
2 - Blue
3 - Green

- Pixeled image

Information → Variance in colour intensity

(Q) Why use another algorithm to solve this

⇒ Image are Special correlation
क्योंकि यहां आस-पास के सारे Pixel आपस में Related होते हैं, तो जो ANN में नहीं हो पाता है।

in ANN → is Structure
x_1 x_2 | x_3 | x_4 | $x_{...}$

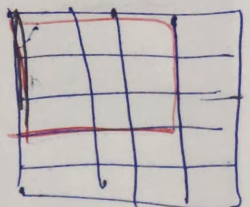
इनको उधार
↑ | x_3 | x_1 | x_2 | x_4 |

} → No Matter but in Image really matter which use.

↑ तो Directly हम लोग flatten नहीं कर सकते image data for use in neural Networks → Do Some special filter.

इसलिए Image data मे पहले filter use करते है which is called Convolution.

eg



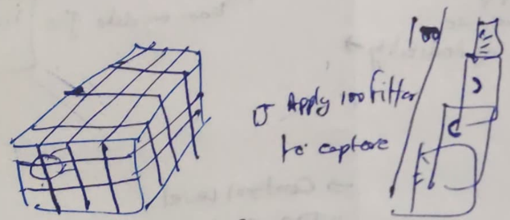
filter

1	1	1
0	0	1
-1	-1	-1

↳ but how to decide this filter?
↓
by Learning from data — Then it is Call - CNN Model

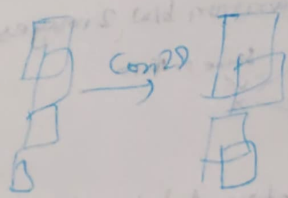
dot Prod
↳ similarity in Two Vectors
If dot product high similar
lower dissimilar

24 sep



Apply 100 filter to capture

→ Con2D



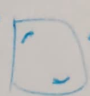
filter to design to only find the Sun image

how by training we do make it

(trend)

more specific

Specific of feature of dog only



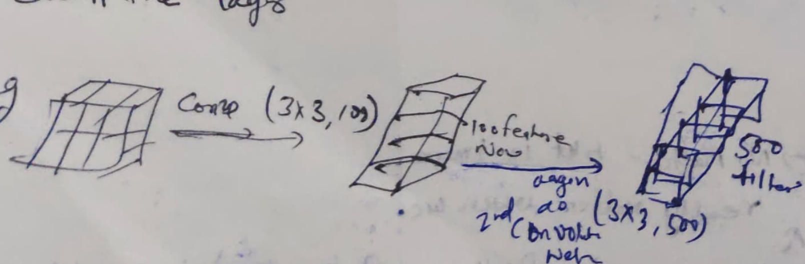
अगर C (dge) Semi Circle
ie mtlb hai to wo dog me
Semi circle का भी extract
करेगा 1 HL layer filter me
तो अब Sun के अलावा आरा
मेरे लिए जरूरी नही है तो और HL
add करेगे Convolution NN
↑
1st layer of CNN

Zigshow Puzzle
↓
only one dog make

Do a cleanup in remove unwanted feat
↳
Use Maxpool 2D

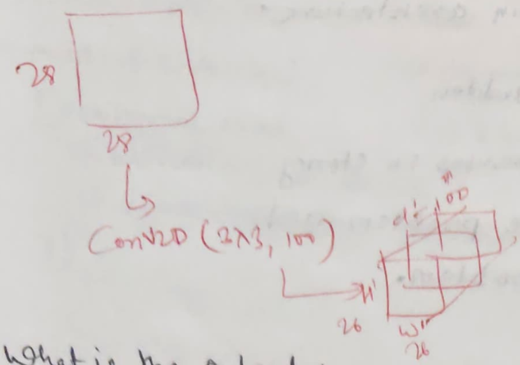
- filter responsibility to extract and clean the layer

eg



- Convonolution extract the feature
- ↳ MaxPooling Clean the feature.

28 sep



Ⓠ What is the output of the Convolutional N in given equation?

Ⓠ What is the no. of trainable parameter?
→ Weight, bias but How
by filter
तो एक filter से कितना
W, B है
3,3 है यहाँ
→ 3×3×3 = 27 weight + 1 bias
28 in one bias
28×100 layers = 2800 W, B

Ⓠ Why we 3 by 3 filter?
→ because if we use bigger filter, it may get unnecessary object in their filter. and it is difficult to extract feature.

Ⓠ Why not 2×2 filter?
→ do not have origin to place → and 3×3 is more uniform to use.

→ MNIST Data set for Practice

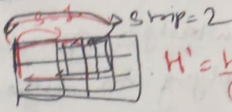
29 sep Transfer learning
→ follow the standard
x_train.reshape(x_train.shape[0], 28, 28, 1)
(No. of image; size; gray)

→ When we convert our image using conv2D and last in flatten then it is called as **Image Embedding**.

→ Here use custom layer

Ⓠ How to access multiple images from folder
→ Colab from glob import glob [illegible]
file = glob('folder/*.png')
for i in file:
img = cv2.imread(i)
cv2.resize(img, (28, 28))
[illegible]
we not guarantee all img same dimension

. Strip → No. of step where Corner will take



$H' = \frac{H - f}{S} + 1$

$W' = \frac{W - f + 1}{S}$

eg. $H' = \frac{28 - 3}{1} + 1$

$= 26$

→ in dense layer reduce Neuron
→ but in CNN increase the Neuron

→ This to avoid flattening use Average Pooling in last before softmax.

Ⓠ How to read raw image for training
→ How to import image.

import cv2
img = cv2.imread("path")
img.shape
>>> img.reshape(1, img.shape[0], [illegible])

25 Sep

Transfer learning.

In Image world

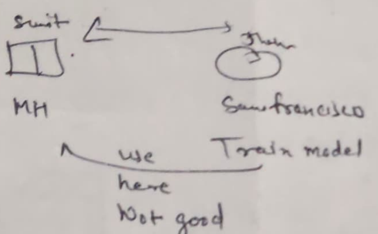
- In Image world create archtechture from scratch it not so much preferable. we use pre train archtecture.
- So why we do that? Why it is change, sudden

It is a technique is ML that focuses on storing knowledge gained while solving one problem and applying to different but related problem.

(OR)

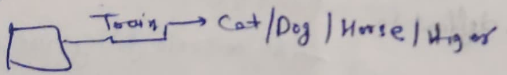
→ million data → imagnet ⟶ trained a archtechture
2010 - 2012-2017

Use case.



but in image are different becase in all local (image has (RGB) combination across.
that use

→ but



- but I use it for only for Cat & dog the How to do it
  if I use it → so I fine tune the train model
  So remove unnecessary layer.
  from trained model I agin fine tuning on that layer called transfer learning.

Some pre train Archtetures are

- Alexnet
- VGG-16
- VGG-19
- Inception net
- Resnet

- So where i find these → in keras Application.
- How I do it
  eg: VGG-16 → 16 → Conv2D
  but before 2014 हमें नही पता था कि ये sequence [illegible]
  but after research [illegible]
  manual भी write [illegible] हो और direct VGG16 भी write कर [illegible]
  [illegible] is not our goal na → Our goal to get weight [illegible] of Pre train arch से → So use keras → [illegible]

1.31 Pre trained_model = tf.keras.Applications.VGG16(weights='imagenet', include_top=False, input_shape=[224,224,3])

Pretrained model.trainable = false

VGG16_model = tf.keras.Sequential([
Pretrained_model,
tf.keras.layers.Flatten(),
tf.keras.layer.Dense(10, activation='softmax')
])

↳ what means → I am not train on weight of VGG16 that I have downloaded

So now My Mode tune by these 2 layer fit for all.

Code run

• Why input_shape optional when include_top is false.

• = यूंकि VGG16 → 5 times max pooling करता है और जो last shape at least 1,1 होना चाहिए अन्यथा minm, input_shape min^m 32×32× ही होगा।

Cosine v_1 v_2 v_2 Projection on v_2 v_1

Oct-5

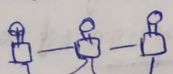
Understanding Image (Embedding).

↳ vector representation

If we convert any data (Video, Audio, Tabular, Image) in 1D Array (embedding) if Convenient

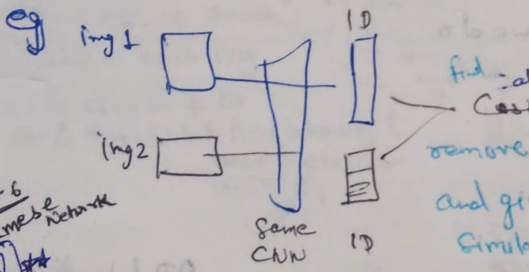
So we can we represent in small represent easier search or so in short.

Resnet



escape connection

Block → It is a arrangement of layer
↳ It is collection of layer
↳ Identity block



find : calculate image similarity
Cosine or euclidean
remove the unnecessary Positional Noise
and give me a both are similar or not img.

→ use in.
↳ Recommendation system
↳ Clustring image
↳ facial Recognition System
↳ siamese Network
↳ image → to text
↳ text to image

Oct 6
Siamese Network

So 1D is very important

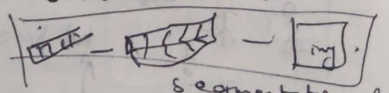
? then How can I get 1D array

↳ Using CNN can → Using CNN pooling → flatten

↳ also Useful to another/or communicate with other data types.
eg: imag to text model

because → image में similars
↳ Closer in image
↳ recommendation image
↳ Product recommendation

Encoder-decoder



segmentation Problem

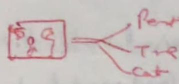
Code

data-generator object
↳ it is pipe of data to we load 1 by 1 or in batch. So it work fine

6 Mar 25

Adv topic

अभी तक we do focus on 1 object only

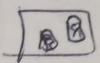


Object dedection And Localization

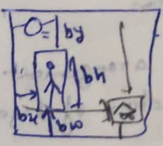
dedect Multiple image + create boundary around them (called localization)

It is dificult Problem to solve
↓
Solution
↓
use pretrain Aroch
- RCNN
- fast RCNN
- faster RCNN
- yolo → yolo 11

- 1st need Supervision
users annotation → How to do it



→ (S) Annotation of bounding boxes to solve for the Supervision of object dedection & localization Problem.

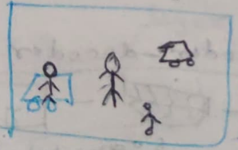


I am Intrest in Human & Car only

kisto do
(1) bx, by, bh, bw, C_{person}^{1}, C_{car}^{0}, C_{Tree}^{0} तो यह Person है
[illegible] Annotated होता है।
(2) bx, by, bh, bw, C_P^0, C_C^1, C_T^0

↓
It is time taking So how we do
↓
by Annotation tools
(1) Robotflow

19 Mar 25
object detection
localiz. Challenge
(8)

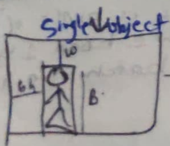


but here output structure in not fixed

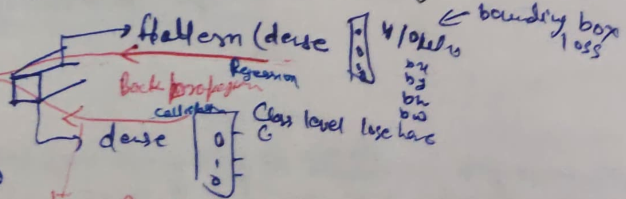
- output structure will alway fixed → How
  we

OD & L में output का structure fixed नहीं है पता चो इससे पहले दे रहे थे कि सब मे output Structure fixed होता है।

- how to solve
  Let Solve the easier problem:
  Can we solve or detect one class with their boundry box



Conv Mp



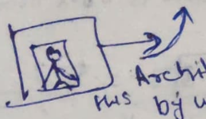
1 img

given img i want detect coordinate also h, w

↓ use multiple img capture
not y

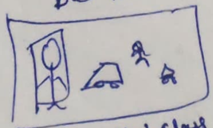
तो इससे कि पता चल सकता है।

→ So by Useing the Code in Class Solve only for sigule image Supervision



this → Architecture by using this solve

↓ but in



अब यहाँ Multiple class हैं तो उसे कैसे single obj में change करेंगे ताकि मे काम कर सके।

Who will tell you? → you can't na. ↓ because you dont no

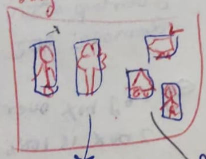
⇓

How to Solve it

(1) 1st Solution → Crop & send to model

but how

So we do by Algos called Region Proposals



Here so based on Property of object create ~~some~~ Clusters on each img

(Here we don't no कि कौन car है, Person है) but Pixel intensity के base में cluster बनाया है।

cluster are called Region Perposal ↗ So based on name यह सिर्फ Perpose कर रहा है, Not sure 100% it it

here 5 perposal here

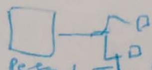
→ use algo to solve by taking confidence.

then जिसमें confidence होगा तो उसमें Boundry draw कर देगा और बाकि में नहीं

So work good

but there is a one problem

↳ It work slow → why because doing 2 thing. So network



Perposal + detection (slow network)

2 Stage Network

why is slow

↳ region Perposal may be some 0 Unnecessary Perposal

What network come into this policy

① RCNN → (It is confuse/ 47sec 1img worst model)

② fast RCNN (2k Perposal)

③ faster RCNN

इसके बाद देखेंगे

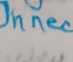
यह 2 stage के बजाय single stage में solve कर सकते हैं → why

use ↓ (1 stage में solve करते हैं)

YOLO (you only look once)

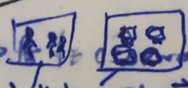
currently use this now

Region Perposal



→ Suggest 'crops' where in you may find an object (Selective Search) Clustering Algos

26th Oct

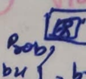


तो कैसे पता चलेगा कि कितना Boundry बनाना है 3, या 4, 5

So it is challege for OD & L.

↓

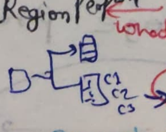
So we solve one at a time



→ by using CNN2D → तो हाँ ये हो सकता है Done solve

Region Perposal अब क्या? ये बस half solve हुआ

What next



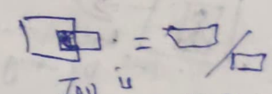
→ objectness score use

Supervison of bonding box

0 means no object of intrest

2000k crops create → but may be 15 crops are useful

Interview Question
↓
IOU (Intrection over union) — before Non Max Suppression
eg ↓
if there is 2 box overlap

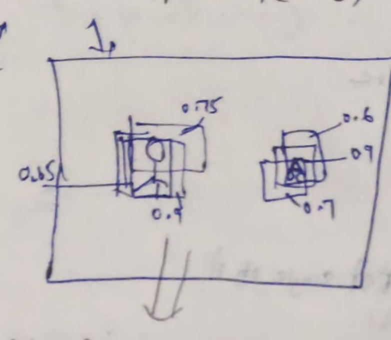


IOU is 50% Value

→ So Not Get all crops we do
⇓
Post Processing — Non-max suppression (NMS)

• what problem we solving?

→ Solve the overlap Crops in Image
⇓
and show the img Crop & boundry region
so do a model the train
⇓



• but we are not assure these boundry work well
⇓
so improve by reduce the/or do a clean up
⇓
by using Non-Max suppress

NMS → for Person class:
• find the box with highest pc (Person cls)
• All boxes having IOU > 0.7 → Threshold we use
with the above box, eliminate → So these are boxes Overlap same with greater than that ये वाली जितना overlap करेगा उसको Remove कर देगे।

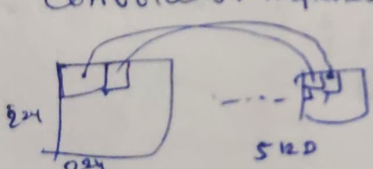
eg: 1 box overlap
2 box is 100% IOU

Called
fast RCNN (All Crop region pass to predict)

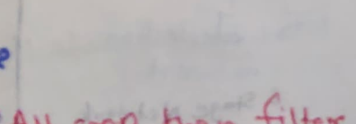
but here also some thing missing and slow.
→ 2 Stage Problem

Faster RCNN (Fast RCNN + RPN) — Region purposed network

Convolution implementation of Sliding Window
यहाँ पर Selectiton Convolution की ही है जाता है।
↳ Key why Learn OD & L Seperately ? हमें तो Architecture होगे।



उसे यहाँ Associate किया है



→ Anchor box
→ but what if we have More object of intrest there lik 5 class then what
↓ so here we use concept Anchor box

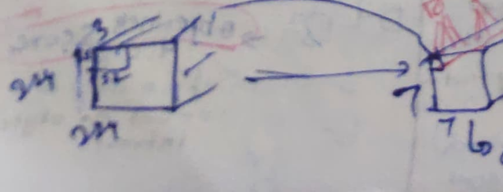
↳ So we change into Supervision

उपर जो Convolution की है तो why we need to Learn OD & L.

⇓
faster RCNN (All crop then filter it eg: 200–14 crop
then 14 crop use for Prediction

Fast OD

• we are doing in 2 stage all → So Same thing do in 1 Stage Process → which is in YOLO (you only Look once)

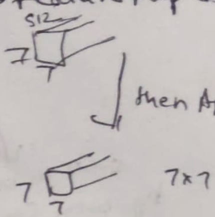


→ but here problem → अगर मेरे पास एक ही 3D में Multiple img उस को कैसे solve करो → Here come Anchor box

↳ Conv (1×1, 8) → $P_c, C_p, C_c, C_7, B_x, B_y, b_h, b_w, b_g$

Anchor box:

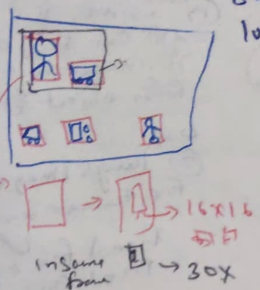
It say put you image → Conv pooling लगाएंगे → feature map. ये आएगी



Eg

तो How Supervision look like

then not dedect only one bounding box
detect multiple boundary box

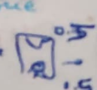


Check objectness score
• है object है या नहीं score.

तो Show करो
$P_c > 0.6$
True

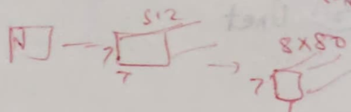
9 NIV YOLO

use YOLO Architecture

but कुछ बार Partially scor  0.5 … 0.5 it good not
तो यह show नहीं करता है

We use model & configure Predefine Architecture

eg:- 80 class हैं

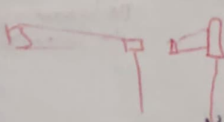


YOLO → combine all data
create cluster
→ & get 9 unique box

जो मिला 16x16
16x32
32x16
32x64
32x32
64x32
64x64, 64x128
128x64

→ तो अब आप Box को Predect करने के लिए ये 9 unique box aspect to solve overlap image.

→ Arch YOLO



why do upsampling here
∴
→ Small feature deal to deficit (?)
1st d. small one feature find
then medium
then large
→ block → Repeating. Model

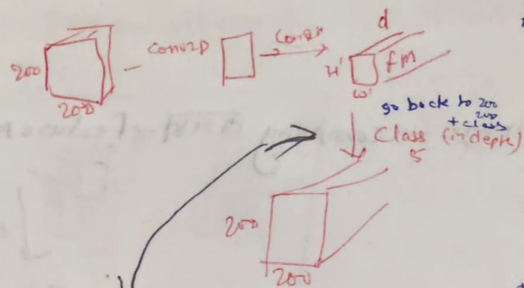
Object Segmentation

→ to dedect execat location → So go with Pixel level.

→ eg Potrate mode or buke effect
or DSLR effect
∘ Medical Image
∘ Multiclass Pixel classification

∘ Find in Pixel level

categorical cross entropy
Adam



Archi tech में Same output pixel होना चाहिए
→ जितने Pixel है उतना ही output निकलेगा
↓
why do
↓

① Road, कहां पे है हमी car self drive में important है सिर्फ object है।

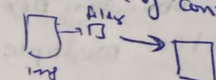
eg.
② lungs image for tumer → region fixed for operation Pixel show

- So Archtecture
  ① FCN (fully convolution net)
  ② Unet

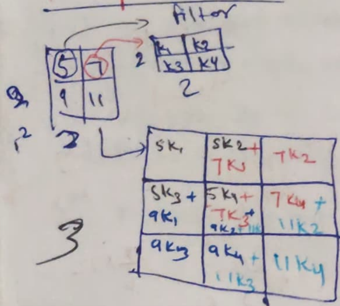
Small to bigger How to go by using Concept

Deconvolution / Transpose Convoletion

↑ reverse of conv

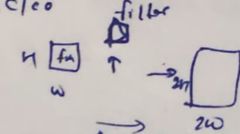


Transpose Convlution



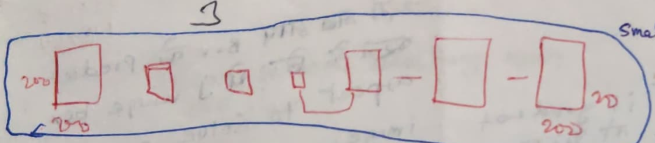
here we data to generate more data from feature, after

TC/eo



→ generative layer
Smaller → bigger data

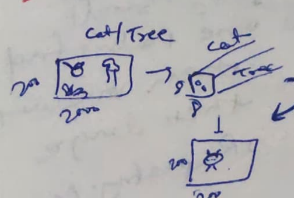
2 option to do
① Upsampling
② Transpose



→ Step in OSeg ?

- Upsampling : (interpolation do)
  no learning here
  do some thing like this का opposite

- Archtecture
  ① Unet



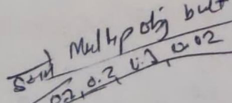
Increase the Scale are if some thing delete

So need some reference Point from Similar layer

① $P_c > 0.6$
② max class proba (class)
③ $P_c > 0.6$ ↑ तभी इसको use करेगा (check)
④ Create the Anchor box
(Bounding Box Achieved)

eg

इसमें Multiple obj but वो किसी भी में confident नहीं है
0.2, 0.3, 0.3, 0.02

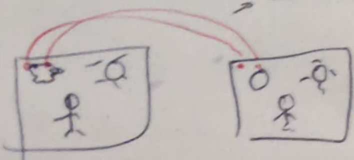


→ Center of Person

Siamese Networks

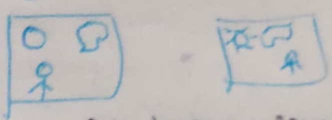
Usecase
- → when compare 2 image same or not.
- • face lock in different location but it work

→ Called Frobenious Norm



distance calculate by Eucleidin

why we do this and what is the problem we face?



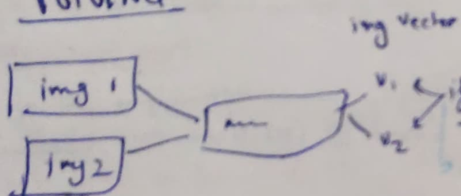
Both image are same but may be some pixel change then it not work. then what we do

→ So we need some feature selection 1st then we go with pixel comparision

eg face

Arch (use Any Arch) eg :- resnet50, vgg...

eg **VGG Net**



- if both img are same then reduce the distance (D²)
- if different then what maximize the distance b/w them. but Avoid $m(1, m-D)^2$ Margin - m

So Loss

$$Loss = \min\left(y_1 D^2 + (1-y_1)\left(\max(0, m-D)\right)^2\right)$$

So D cross M many for better max details

↳ which is called

Anchor Loss

Code

01:00

→ After this go to GAN
Generative Adversarial Network
eg Img → [Model] ← new img
Able to Generate new

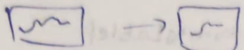
GAN (Generative Adversal Network)

img → [model] ⇐ generate new Img

Use case

1. Increase the data set when we have Less data
2. Nural style transfer

   eg: I have panting → & I want make my Painting in Delenau. Style there.



eg This is Delanutes Paint → I understand it & apply on my point

(good for content Createar but where is business Use case)

3. GANs with NLP (Generating image from text)

• 1st Understand Concept

Generater

Noise (Random number distribution) called Z array by → [Generate model] using (ANN) → o/p → $G(z)$

→ size (flatten) 1D array of $r*c$ → may be Real/fake → go to

1D $[a_{11}, a_{12}, a_{21}, a_{22}]$ in (NN only in 1D) & img in 2D so

$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$ ← Then Reconstruct

↓ maxinize $D(G(z))$

Discriminator

[Discriminator Model]

↳ $D(x)$ → Real or fake (binary)

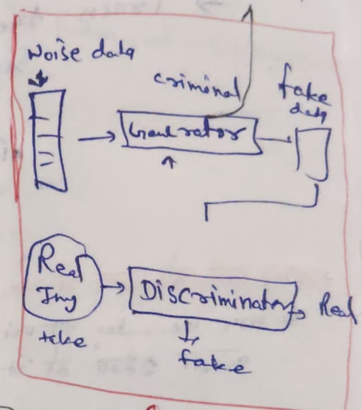
binary classification

$D(G(z))$

↓ Maximize

$D(x)$ → Score of Real img

$D(G(z))$ → Less prob of Real to fake images



update — Binary Real or Not

train together Called generator & Discriminator network or also called GANs

↳ How Loss function work

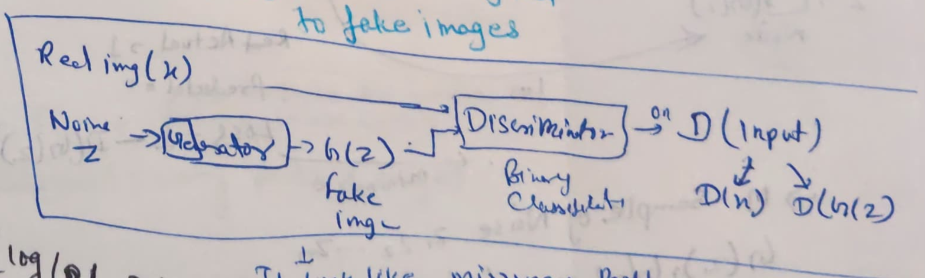
Generator Not No How Real data look it just

Objective of GAN Model

↳ $\min_{G} \max_{D} V(G, D)$

V = loss
G = Generator
D = Discriminator

$= E(\log(D(x)) + E(\log(1 - D(G(z))))$



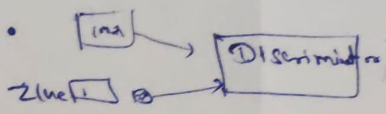
It look like minimax Problem

⇒ So we Train first

Training GAN Model :-

1. Training Discriminator :-

Pre Actual Loss
0.3 -1 0.7 ← improve in Discrim
0.35 -1 0.65 →



m sample of real image $x_1, x_2 \cdots x_m \rightarrow$ [1, 1, 1 ...] O/P binary Label
n " " " noise $z_1, z_2 \cdots z_m$ [0, 0, ... 0]

→ Make generator non-trainable (θ_G)
→ Train the discriminator
Sure (S G D)

$$\frac{d}{d\theta} \left[-\frac{1}{m} \sum_{i=1}^{m} \left[\log(D(x_i)) + \log(1 - D(G(z_i))) \right] \right]$$

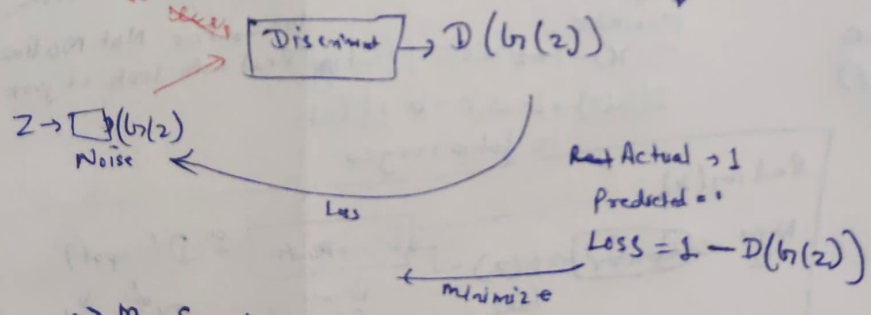
- use to decrease the positive thing's
So Maximize in G D

- Binary Cross Entropy Loss
θ_G is Random

अगर दोनों को साथ में train करते तो क्या issue होता?
↳ अगर Generator भी उसी time generate होगा और D check करता रहेगा तो it is never ending loop में जाएगा।

2. Training Generator.

~~Random~~ no need Random



→ m Sample of Noise $z_1, z_2 \cdots z_m$
$G(z_1), G(z_2) \cdots G(z_m)$
→ Make discriminator (θ_D) non trainable
→ Train generator to Update θ_G using S G D

$$\frac{d}{d\theta_G} \frac{1}{m} \left[\sum_{i=1}^{m} \log\left(1 - D(D(z_i))\right) \right]$$

Steps of training GAN:

④ Train & test
⑤ optimize
⑥ when to stop

→ Learn the objective of Problem
→ Decide the architecture of Gen & Disc
↳ For 'n' epochs (1 train data cycle)
↳ Train disc θ_G θ_D
↳ Train gen

→ Arch of Gen & Disc
→① NN with 3 Hidden layer, each followed by LeakyRelu (non-linearity) / or Dropout (regularization)
② Output layer
↳ Gen → τ_{a+c} → tanh use [-1, 1] → (1 to 255 pixel value normalize then to [-1, 1])
↳ Disc → Sigmoid (0,1) Prob

- Adagard → Optimizer

24 Nov

Special types of GAN

- DEEP Convolutional GAN (DCGAN)

use case:

- Image ~~Re~~ High Resolution Image (use Pretrain to reduce the business cost)
- Image Inpainting (Reconstruct missing parts of images)
- Art & Design.

Strength

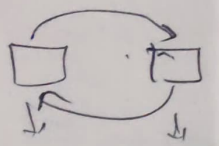
- High Quality Img
- Leverages CNN for better spatial feature learning
- Relative simple to implement & train.

Challenges:

- Still prone to mode Collapse (generator outputs limited variation).
- Sensitive to hyperparameter (eg lr)
- Struggles with very High Resolution image or complex data

② Cyclic GAN

- use 2 Generator 2 Descriminator



$G_1 : x \to y$ $\quad F : y \to x$

Produce

$G_1 : x \to [G_1] \to y$

$F : y \to [G_2 / F] \to x$

$D_x : \to [D_x] \to$ real of x Prob

$D_y : \to [D_y] \to$ y होने का Real Prob का आउटपुट

Objective function

$$[\log (D_x(x))) + E[\log$$

+

Cycle Loss

$= E($