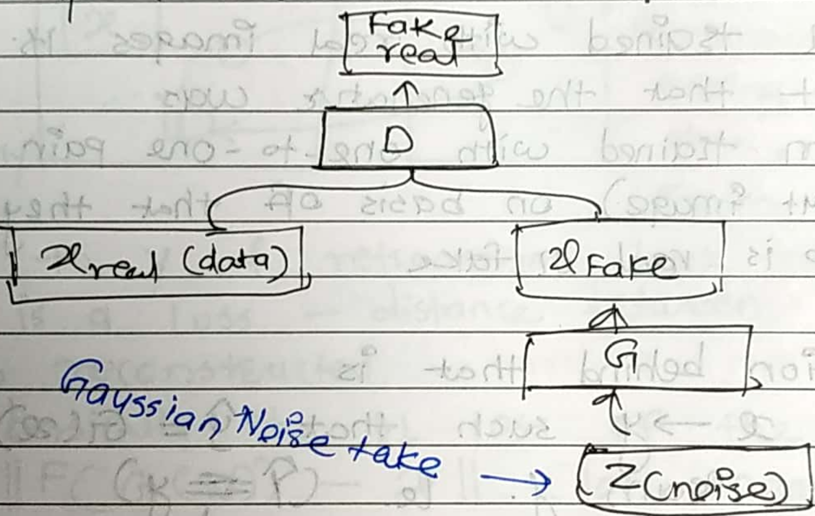
GAN

Generative Adversarial Netw

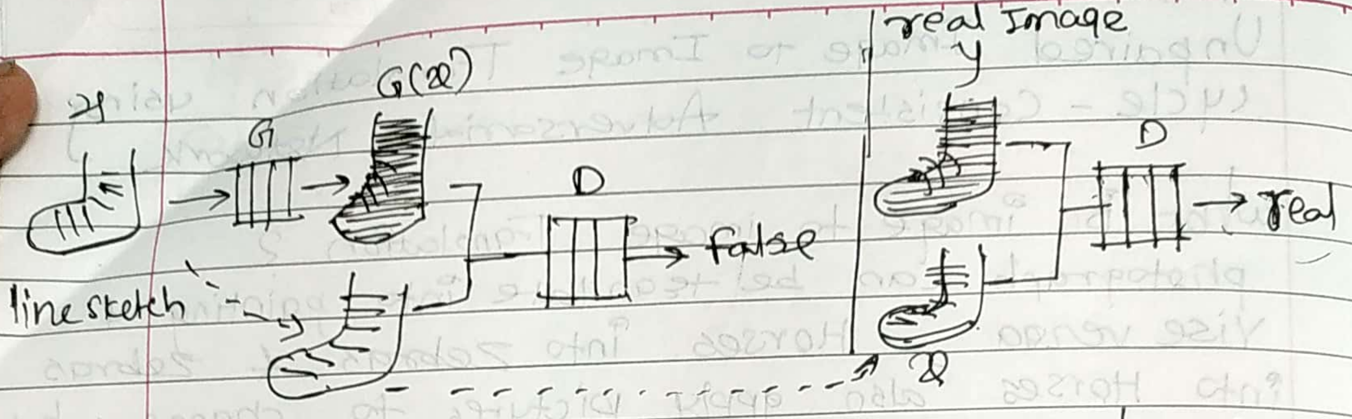
Page No.
Date

Unpaired Image to Image Translation from
cycle - Consistent Adversarial Networks you
uld be

What is image to image Translation ?
photograph can be translate into painting &
vise versa , Horses into zebras & zebras
into Horses also apply pictures to change what
the seasons look like. (winter → summer) &
(summer → winter) Also
Take area of image & translate into Google
map & also do reverse.



Gan takes a Noise inputs into Generator
Create something that's feed to a discriminator &
~~that do discriminator~~ also we have real data
so this discriminator here tries to differentiate
between this two and the feedback goes
back to generator better at creating fake
images & also feedback to discriminator to
make better at differentiating those & so
that's the adversarial Nature of these two
things that are pretty good at generating
some results



(1) Sketch goes into input & generate photograph & discriminator is along with the original image provide to it so that this helps it makes its decisions & ~~then~~

(2) Discriminator trained with real images its provide the same input that the generator was.
Discriminator trained with one-to-one pair of (real img : input image) on basis of that they said Generator o/p is real or fake

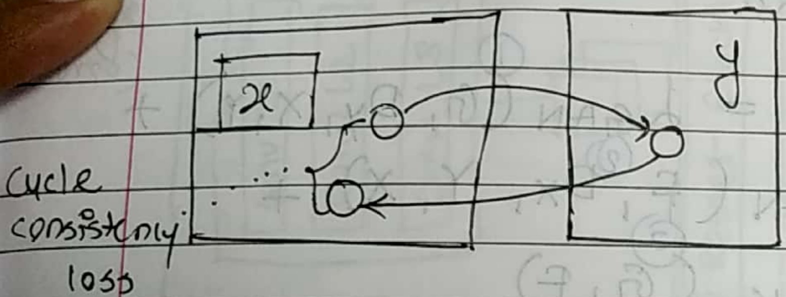
The intuition behind that is
Train $G: x \rightarrow y$ such that $\hat{y} = G(x)$ is indistinguishable from y. i.e. $(\hat{y} \approx y)$
that's what the gan can do, we can start mimic that distributions as we continue to train it.

but problem is that we are not guaranteed any meaningful correspondence between the input & the O/P

So we come up with the idea of cycle consistency
The idea is that if you have
Given $G: x \rightarrow y$ & $F: y \rightarrow x$,
then $F(G(x)) \approx x$ & $G(F(y)) \approx y$

If your G transformation that takes you from x to y & F transformation that takes you from the y back to x then you should be able to take your image
Ex - horse → zebra : G &
zebra → horse : F then
ie. input x → Generated G(x) → Reconstruction F(G(x))

Cycle Loss Formulation

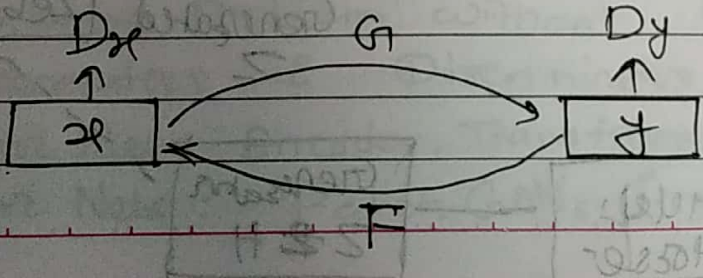


-Gan cycle loss formulation basically is taking this idea again that, we start from x domain Tx to y & retransform back to x measure this is a loss - distance between those once its been reconstructed.

basically the L1 loss of the reconstructed image $\| F(G(x)) - x \|$ loss measured & vise versa

$$L_{cyc}(G, F) = E_x(x) \left[\| F(G(x)) - x \| \right] + E_y(y) \left[\| G(F(x)) - y \| \right]$$

Full Objective
GAN Loss Formulation



Page No.
Date

x: class 1 (Horses)
y: class 2 (Zebras)
D_x : Generator O/P
D_y : Discriminator O/P

$$\mathcal{L}_{GAN}(G, D_Y, X, Y) = E(y)\left[\log D_y(y)\right] + E(x)\left[\log(1 - D_y(G(x)))\right]$$

Full objective

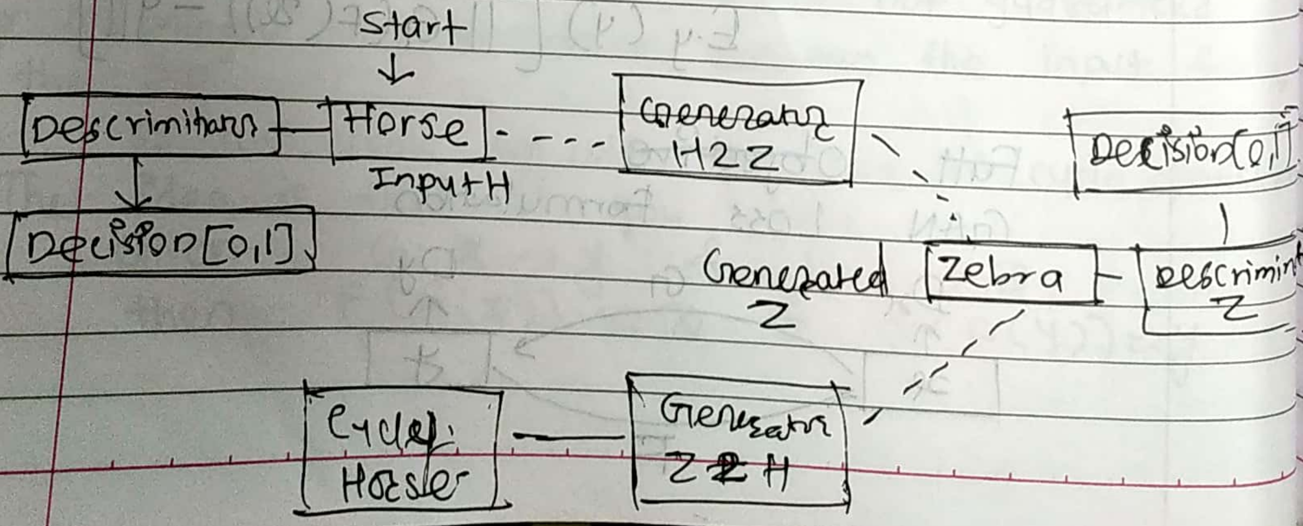
$$\mathcal{L}(G, F, D_x, D_y) = \mathcal{L}_{GAN}(G, D_Y, X, Y) + \mathcal{L}_{GAN}(F, D_X, Y, X) + \lambda \mathcal{L}_{cyc}(G, F)$$

1. Gan loss for Horses → Zebras
2. Gan loss for Zebras → Horses
3. Cyclic loss

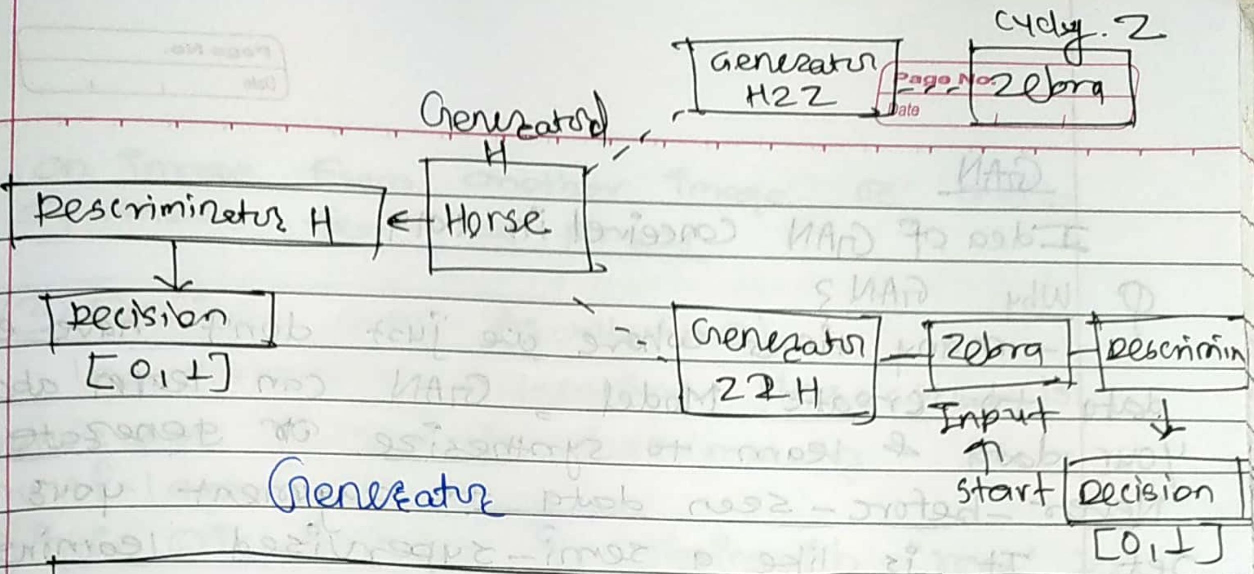
$\boxed{\begin{array}{l} \text{Min} - G \\ \text{Max} - D \end{array}}$

$$G^*, F^* = \arg \min_{G,F} \max_{D_x, D_y} \mathcal{L}(G, F, D_x, D_y)$$

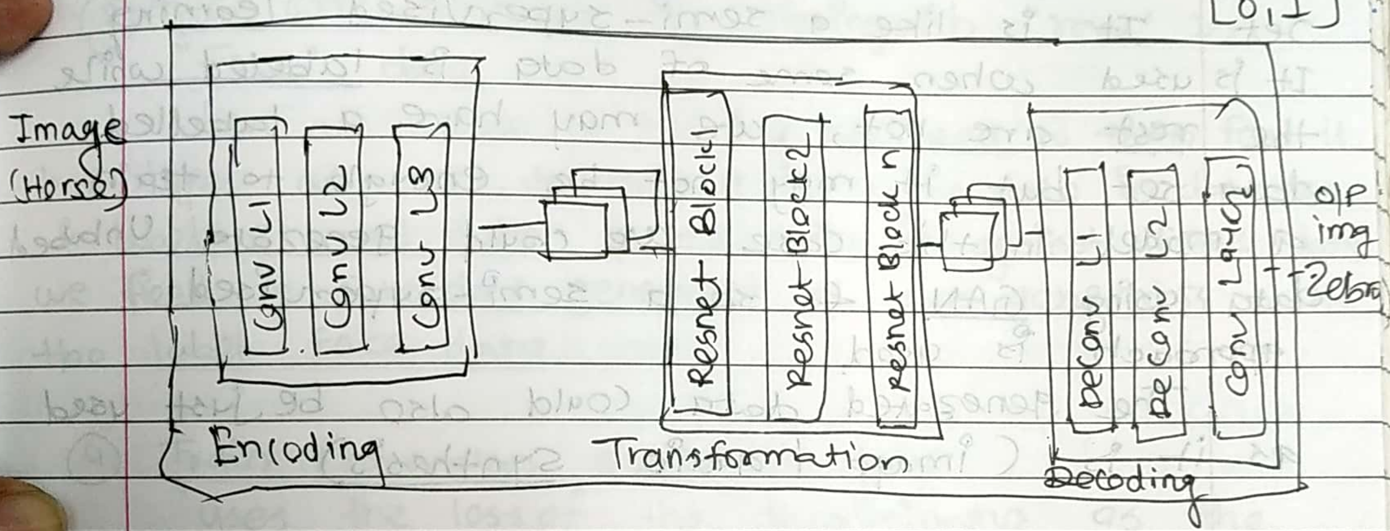
N/w



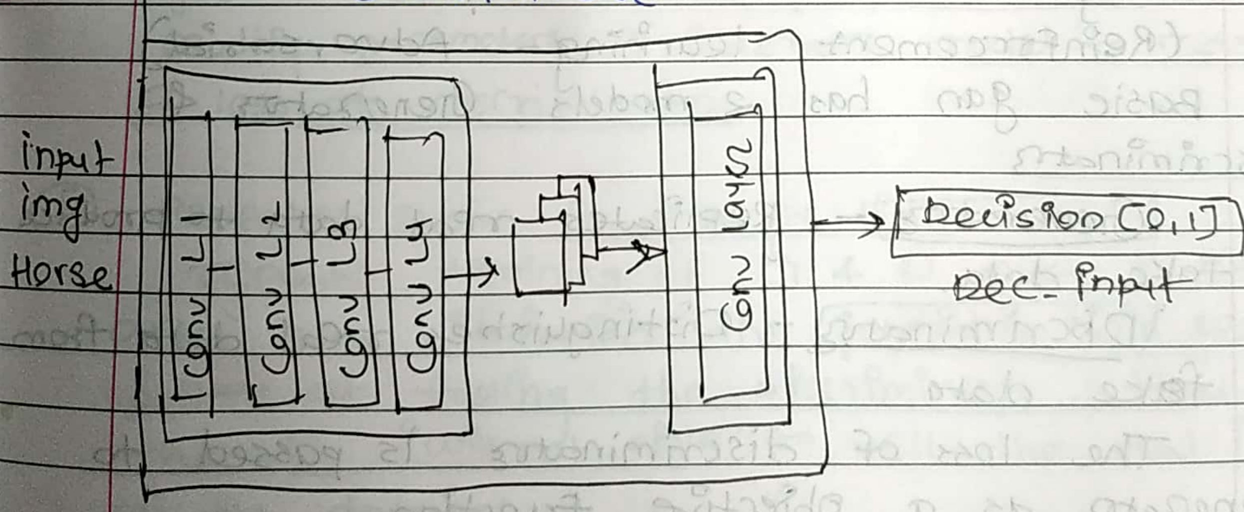
Cycle. 2



Generator



Discriminator



Cycle Consistency Adversarial Nets - solves image-image translation problem without paired data.

- 2 Generators & Discriminator Networks

Generator Net : Encoder, Transformer, & Decoder

Discriminator Nets : Patch GANs implemented as FCN

Page No.
Date

GAN

Idea of GAN conceived in 2014

Q. Why GAN?
- many times where we just don't have enough data to create Model, GAN can learn about your data & learn to synthesize or generate Never-before-seen data to augment your data set. It is like a semi-supervised learning. It is used when some of data is labeled while the rest are not, we may have a labelled data set but it may not be enough to train a model in this case we could generate Unlabeled data using GAN & so a semi-supervised approach is used.
The generated data could also be just used as it is (image / audio synthesis)

Q why is it Adversarial?
(Reinforcement learning - Adversarial)
Basic gan has 2 models Generator & descriminator
$\boxed{\text{Generator}}$ - Replicates real data to produce fake data.
$\boxed{\text{Discriminator}}$ - Distinguishes real data from fake data
The loss of discriminator is passed to Generator as a objective function

Steps to Train a GAN
① Define the problem
what trying to do to synthesize images from a caption or do we need synthesize

an image from another image or audio synthesis from sentences

② Define GAN Architecture
i.e. Answering questions like what is Generator, what is discriminator. It is a multi-layer NN

③ Train Discriminator to distinguish "real" Vs "Fake" data
when we speak to training we need to ~~train~~ feed it both types we need to train it from the data in dataset & label as real data & to train it against fake data we feed it fake data generated by our generator with the label fake data

④ Train Generator synthesize data
uses the loss of the discriminator as the objective function to the generator, we need to modify parameters of Generator model to maximize a loss of discriminator

⑤ Repeat steps 3 & 4 N time
repeate training of G & D over n epoches after every iteration the generator will get better at fooling the discriminator, finally the D will not able to tell the real images of the dataset,
once training is complete, synthesize data from Generator & this can be used to augument or true data set & use as it is because it is really cool (Real).

Loss function

- Discriminator uses Cross-Entropy loss. why?
- it's because that we are dealing with a Classification model, Cross Entropy loss is a better performance metrics than classification error or mean squared error MSE (Misclassification Rate)
- Misclassification rate fails to state how wrong / how correct predictions are
  In cross-entropy loss it is possible to have a model which miss classify samples more as a better model (when it was wrong it wasn't really wrong by that much)
- Other reasons for cross-entropy is to Eliminate vanishing gradient in NN Architecture (change in weight does not become zero when using cross-entropy)

Cross entropy loss between p & q

p - true distribution

q - Estimated distribution

$$H(p,q) = -\sum_i p_i \log(q_i)$$

$p \in R^M \quad q \in R^M$

p & q are vectors of M diamensions where M - No. of classes

Now given sample can only belong to a single class so True distribution is one-Hot vector with one 1 & rest zeros (i/p sample belong to certain class)

- Descriminator in GAN is binary classifier (real or fake) ie $M = 2$

Page No.
Date

ie. one hot encoder with consisting of two terms

$$H(p,q) = -p_1 \log(q_1) - p_0 (\log p_0)$$

$$H(p,q) = -(1-p_1)\log(1-q_1) - p_1 \log(q_1)$$

for N samples, cross entropy loss

$$H(p,q) = -\sum_{i=1}^{N}(1-p_{i,1})\log(1-q_{i,1}) - \sum_{i=1}^{N} p_{i,1}\log(q_{i,1})$$

we know that half sample come from the data & other half from generator

Half of the sample that follow the distribution of real dataset is $x \sim p_{data}$

Half sampled from fake generator data $x \sim p_{generator}$

$\sim$ - destributed as

$D(x)$ - predicted output of Discriminator

$$H(x,D) = -\frac{1}{2}E_{x\sim p_{data}}[D(x)] - \frac{1}{2}E_{x\sim p_{generator}}[1 - D(x)]$$

$$E_{x\sim p_{generator}}[D(x)] = 0$$

Expectation of $D(x)$ when x is sampled from the Generator

$$E_{x\sim p_{data}}[1 - D(x)] = 0$$

Expectation from $1 - D(x)$ when x is sampled from real data $= 0$

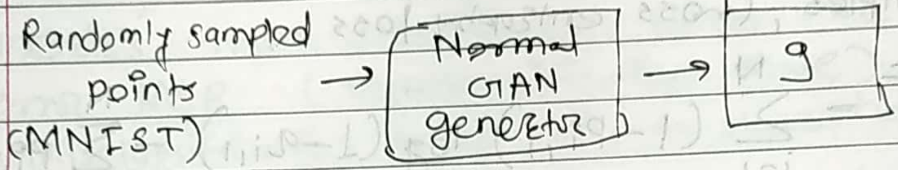
Types of GAN

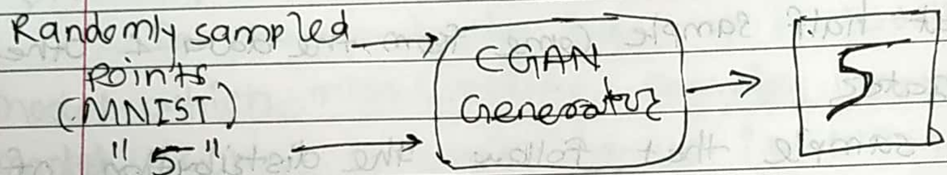
(1) Deep convolutional GAN (DCGAN)

- CNN used in Unsupervised learning
- by using CNN G & D
- (Conv + Deconv) layers

② Conditional GAN

~~Dictect~~ Dictate that the type of data should be generate through CNN using conditions. ie. Conditional GAN





ie. we can direct the generator to synthesize the specific images

③ Info GAN

-Not only Generates images but also additionally learns latent variables without labels in the data

④ Wasserstein GAN (WGAN)

problem with GAN & DCGAN is objective functions, recall the object fun is to increase the loss of the discriminant but their is no clear sign of when to stop, you need to keep looking samples & see if there are satisfactory enough to pass real data

-Replace the old method of measuring loss, Jenson-Shannon divergence with the new wasserstein distance. It is methed to measure the similarity betwn two probability distribution

- using that we can train discriminator until convergence, leading to higher quality samples generated by generator

⑤ Attention GAN
- Text to image synthesis
- latest developed by microsoft Attension GAN
- fine-tuned to draw parts of image from a single word in sentence
- uses a Attention Mechanism: generates parts of img in high resolution & others in low resolution.

Generative Vs Discriminative Algo

Discriminative Algo map features to labels., one way to think about generative Algo is that they do the ~~otr~~. opposite. Instead of predicting a label given certain features they attempt to predict features given a certain labels.

GAN -
Genarative - Generate