7:01 PM Thu Jan 31 96%
Notebook (6) application_form_IIITNR Attachments
Ahlad Kumar

GENERATIVE ADVERSARIAL NETWORK

(GAN)

Conflict or opposition

GAN are deep neural net architectures comprised of two neural networks, competing one against the other (i.e why adeversarial)

7:29 PM Thu Jan 31
Notebook (6) | application_form_IIITNR | Attachments | Notebook (6)
Ahlad Kumar

→ GAN are neural networks that are trained in an adversarial manner to generate data mimicking some distribution

→ Two classes of models in machine learning

(a) Discriminative model : It is the one that discriminates between two different classes of data.

(b) Generative model : A generative model G to be trained on training data X sampled from some true distribution D is the one which, given some standard random distribution

7:30 PM Thu Jan 31
Deep Learning 27: (1) Generative Adversarial Network (GAN): Introduction and Back-Propagation
Notebook (6) | application_form_IIITNR | Attachments | Notebook (6)
Ahlad Kumar

→ GAN are neural networks that are trained in an adversarial manner to generate data mimicking some distribution

→ Two classes of models in machine learning

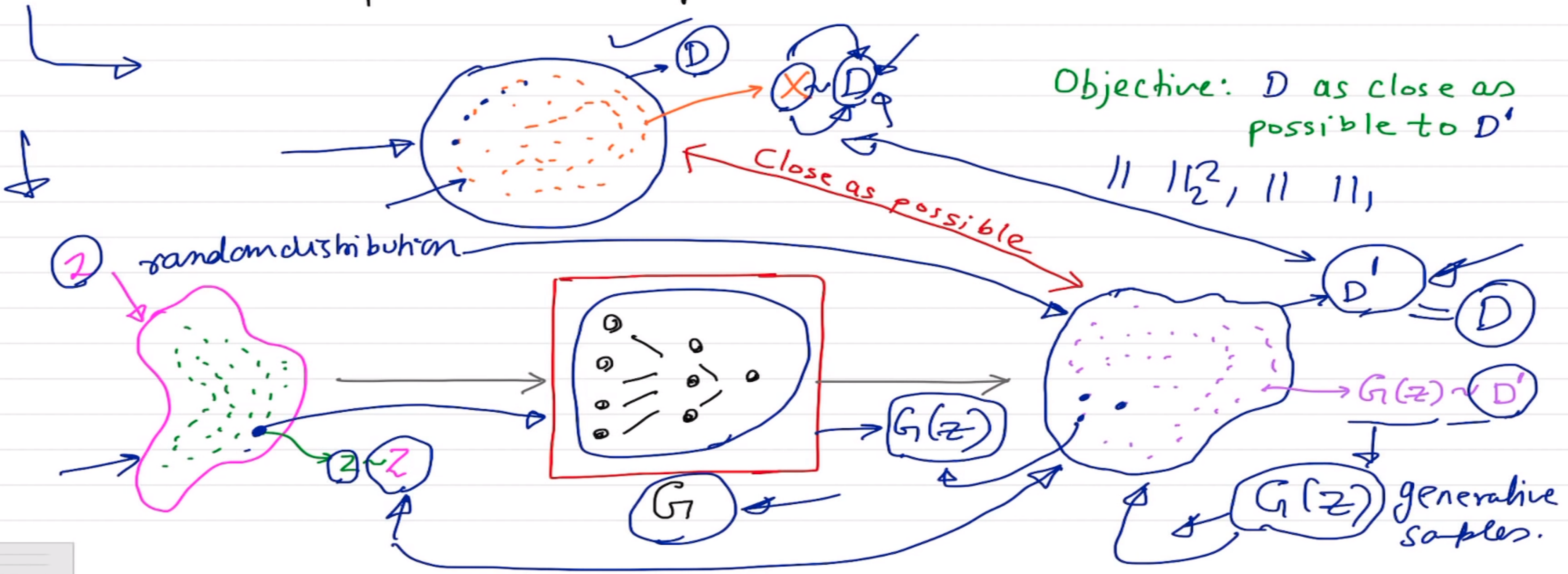
(a) Discriminative model: It is the one that discriminates between two different classes of data. → classification problems $\left.\begin{matrix}0\\1\end{matrix}\right\}$

(b) Generative model: A generative model G to be trained on training data X sampled from some true distribution D is the one which, given some standard random distribution

4:29 / 19:17

Z produces a distribution D' which is close to D according to some closeness metric. Mathematically,

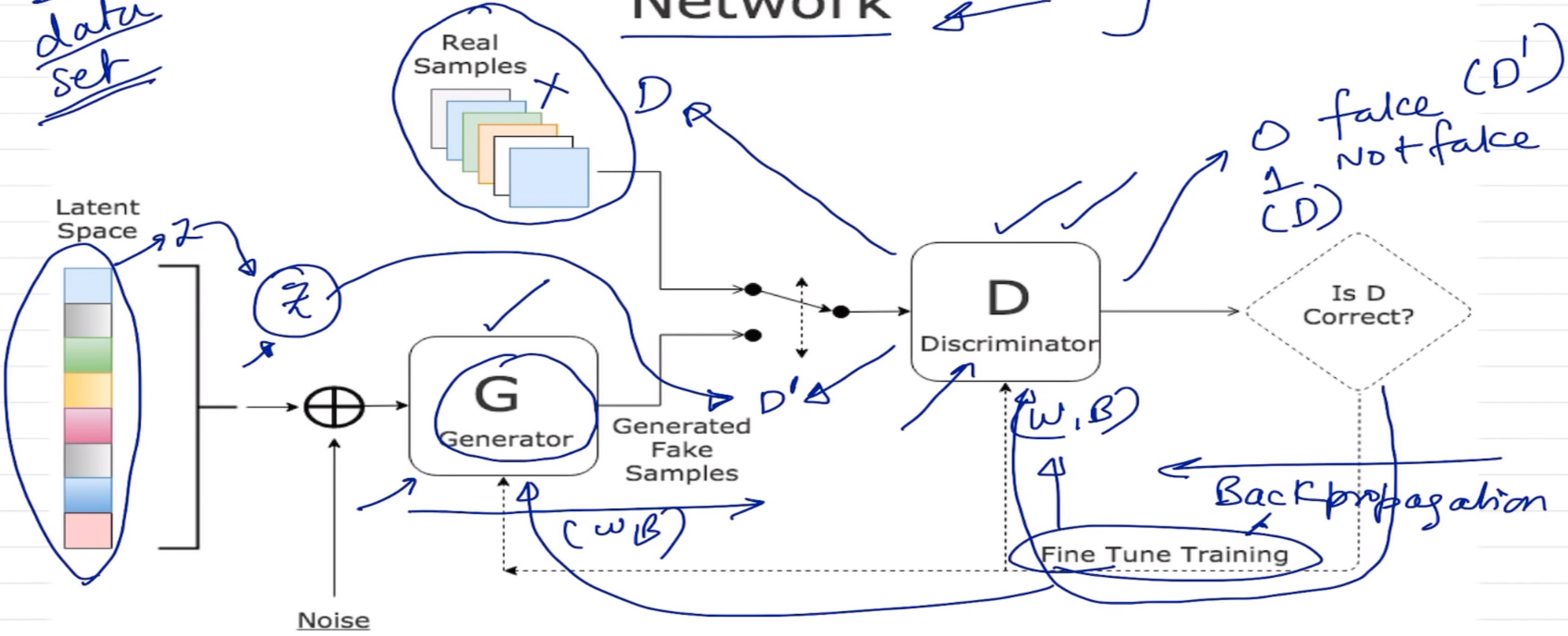
$z \sim Z$ maps to a sample $G(z) \sim D'$



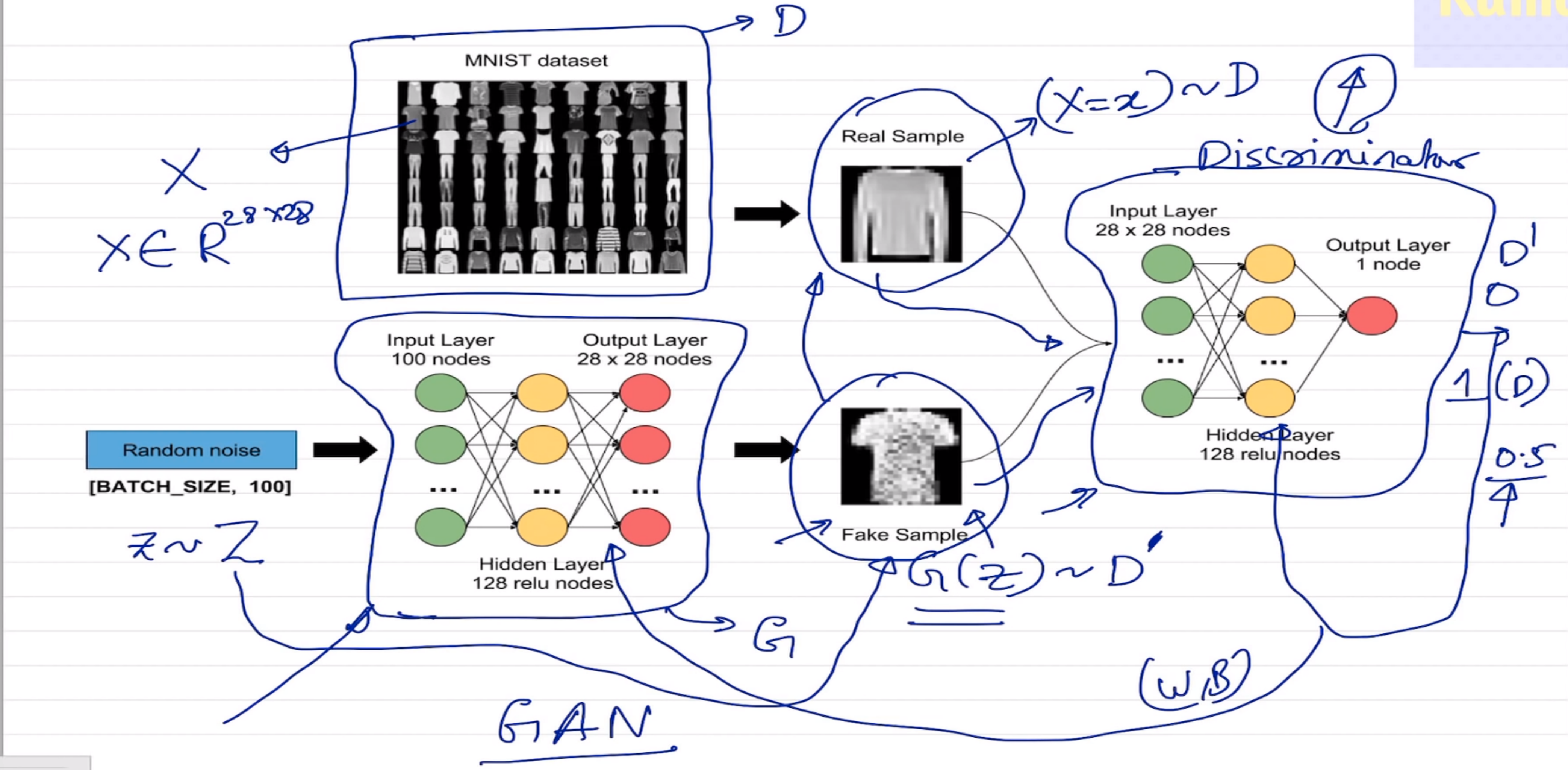
Ahlad Kumar

MNIT Fashion data set

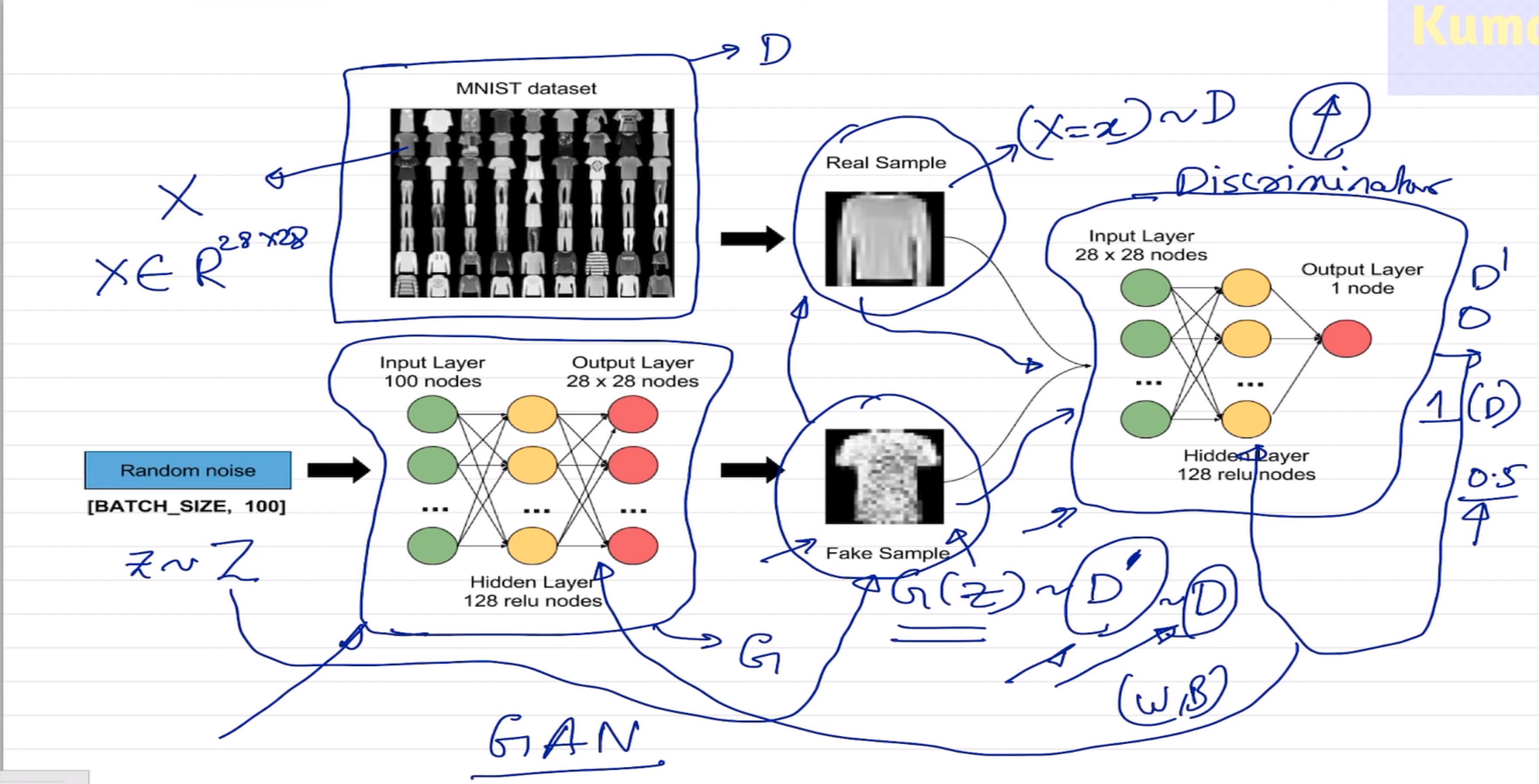
Generative Adversarial Network



8:26 PM Thu Jan 31 81%
Notebook (6) application_form_IIITNR Attachments Notebook (6)
Ahlad Kumar



Ahlad Kumar

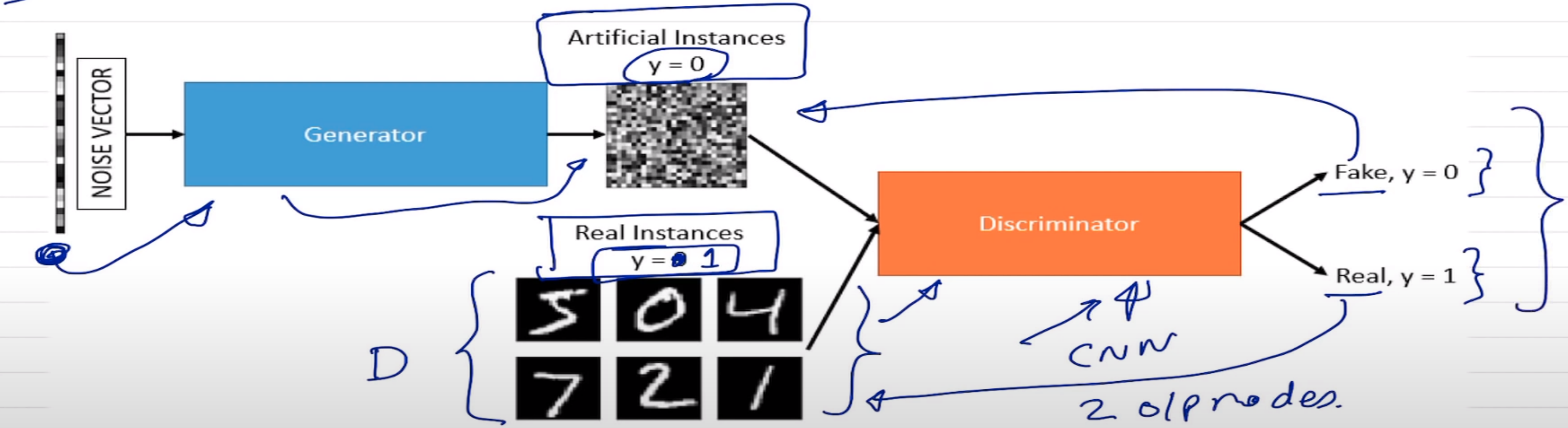

D
MNIST dataset
X
X ∈ R^{28×28}
Real Sample
(X=x) ~ D
Discriminator
Input Layer
28 x 28 nodes
Output Layer
1 node
Hidden Layer
128 relu nodes
D'
0
1 (D)
0.5
Random noise
[BATCH_SIZE, 100]
z ~ Z
Input Layer
100 nodes
Output Layer
28 x 28 nodes
Hidden Layer
128 relu nodes
G
Fake Sample
G(z) ~ D' ~ D
(W, B)
GAN

8:42 PM Thu Jan 31 78%
Deep Learning 27: (1) Generative Adversarial Network (GAN): Introduction and Back-Propagation
Notebook (6) | application_form_IIITNR | Attachments | Notebook (6)
Ahlad Kumar

Learning Mechanism (W, B)

Backprop (D, G)

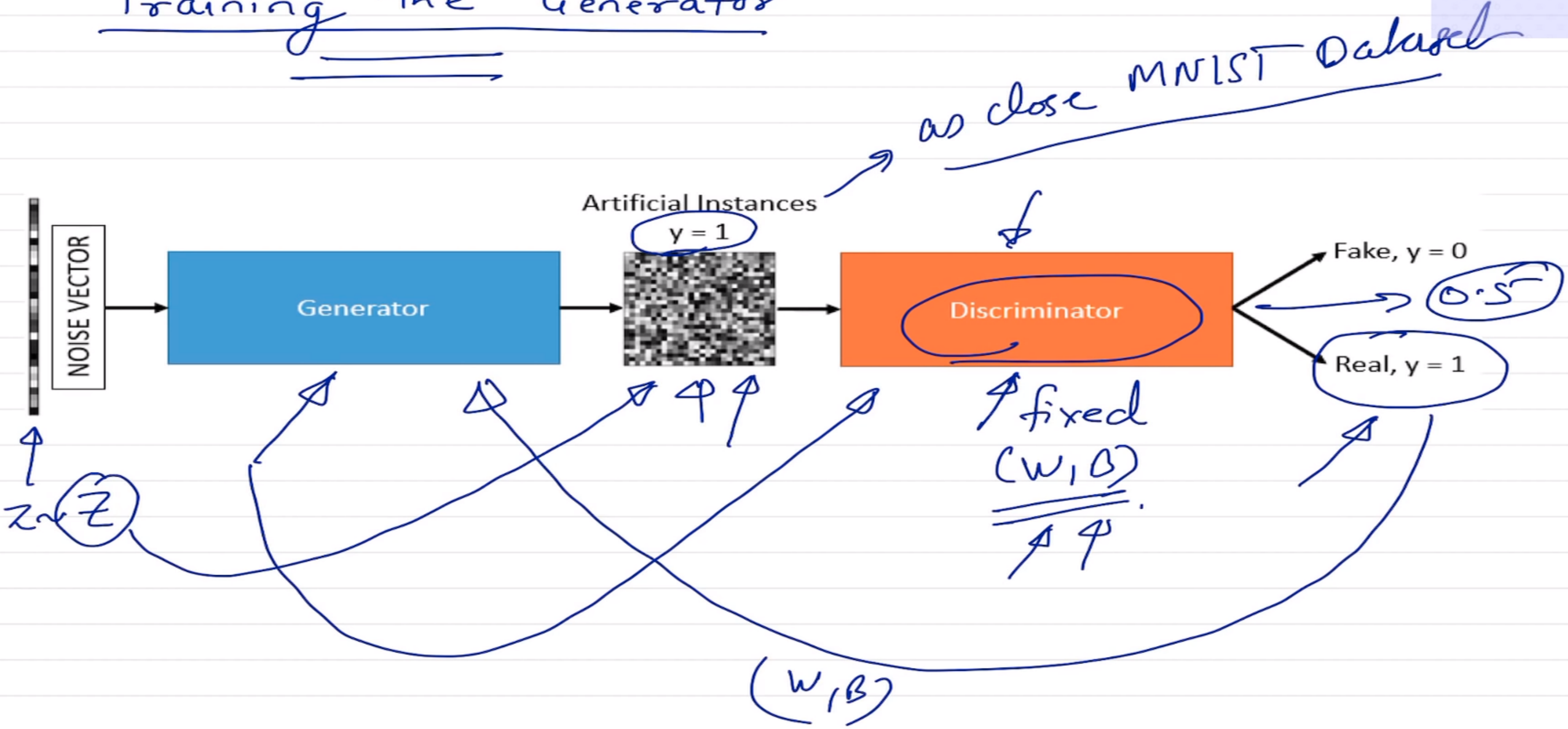
Training the Discriminator



16:54 / 19:17

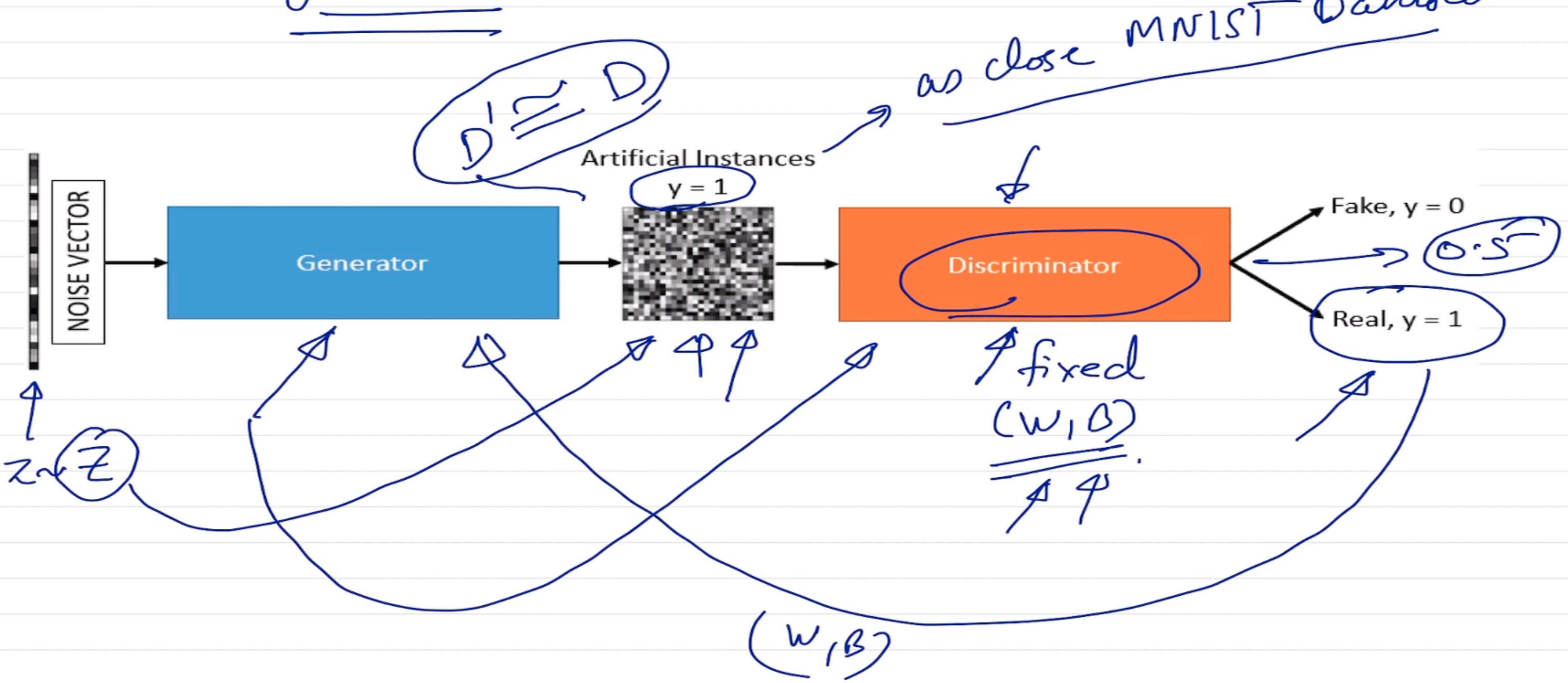
8:45 PM Thu Jan 31 77%
Notebook (6) | application_form_IIITNR | Attachments | Notebook (6)
Ahlad Kumar

Training the Generator



8:45 PM Thu Jan 31
Notebook (6) application_form_IIITNR Attachments Notebook (6)
Ahlad Kumar

Training the Generator



10:39 AM Fri Feb 1
Deep Learning 28: (2) Generative Adversarial Network (GAN) : Loss Derivation from Scratch
Notebook (6) application_form_IIITNR Attachments Notebook (7) Notebook (8)
Ahlad Kumar

→ Loss Function of GAN

- Discriminator : Role is to distinguish between actual data & fake data

- Generator : Role is to create data in such a way so that it can fool the generator.

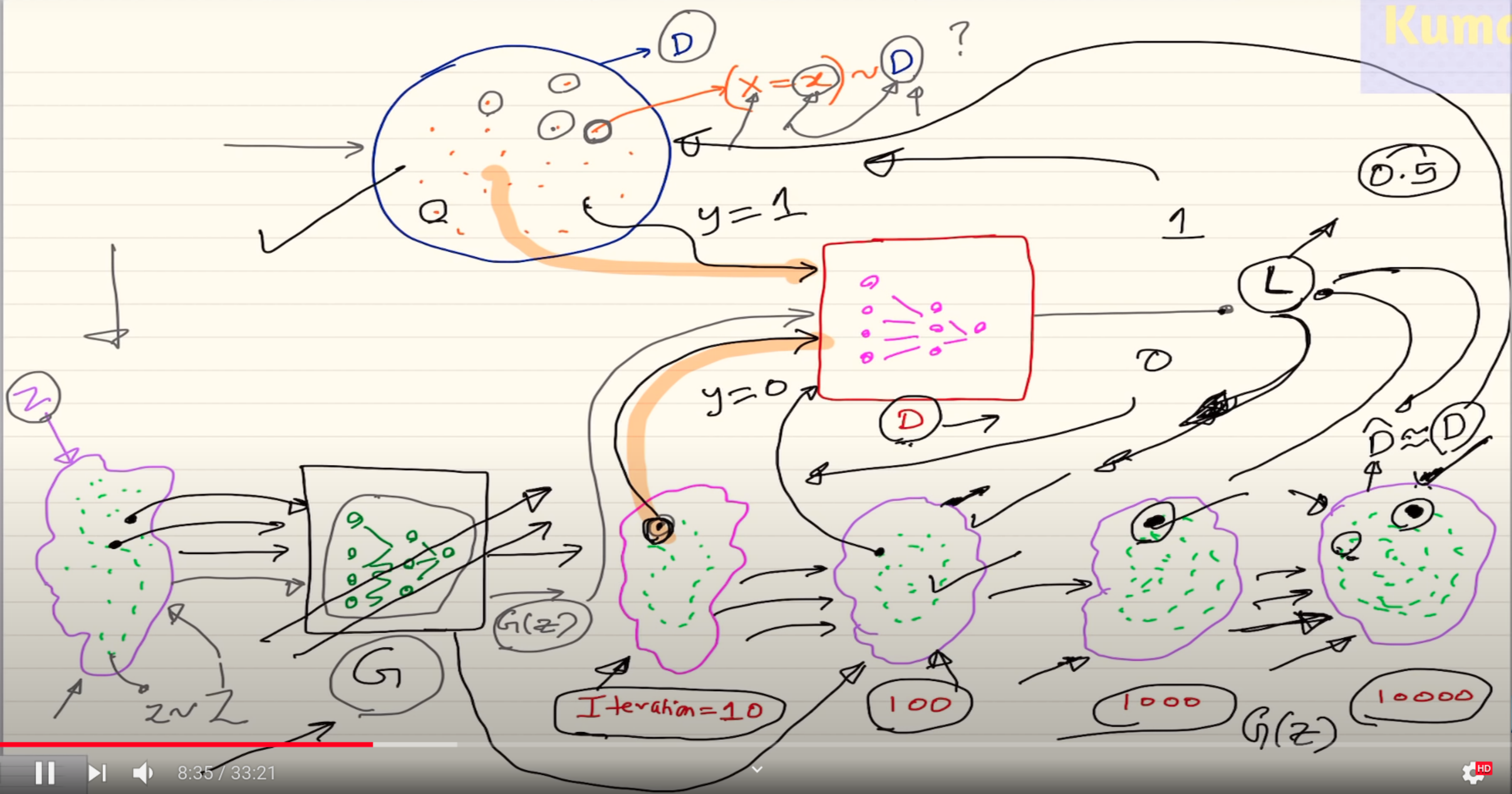
1:06 / 33:21

10:48 AM Fri Feb 1

Deep Learning 28: (2) Generative Adversarial Network (GAN) : Loss Derivation from Scratch

Notebook (6) | application_form_IIITNR | Attachments | Notebook (7) | Notebook (8)

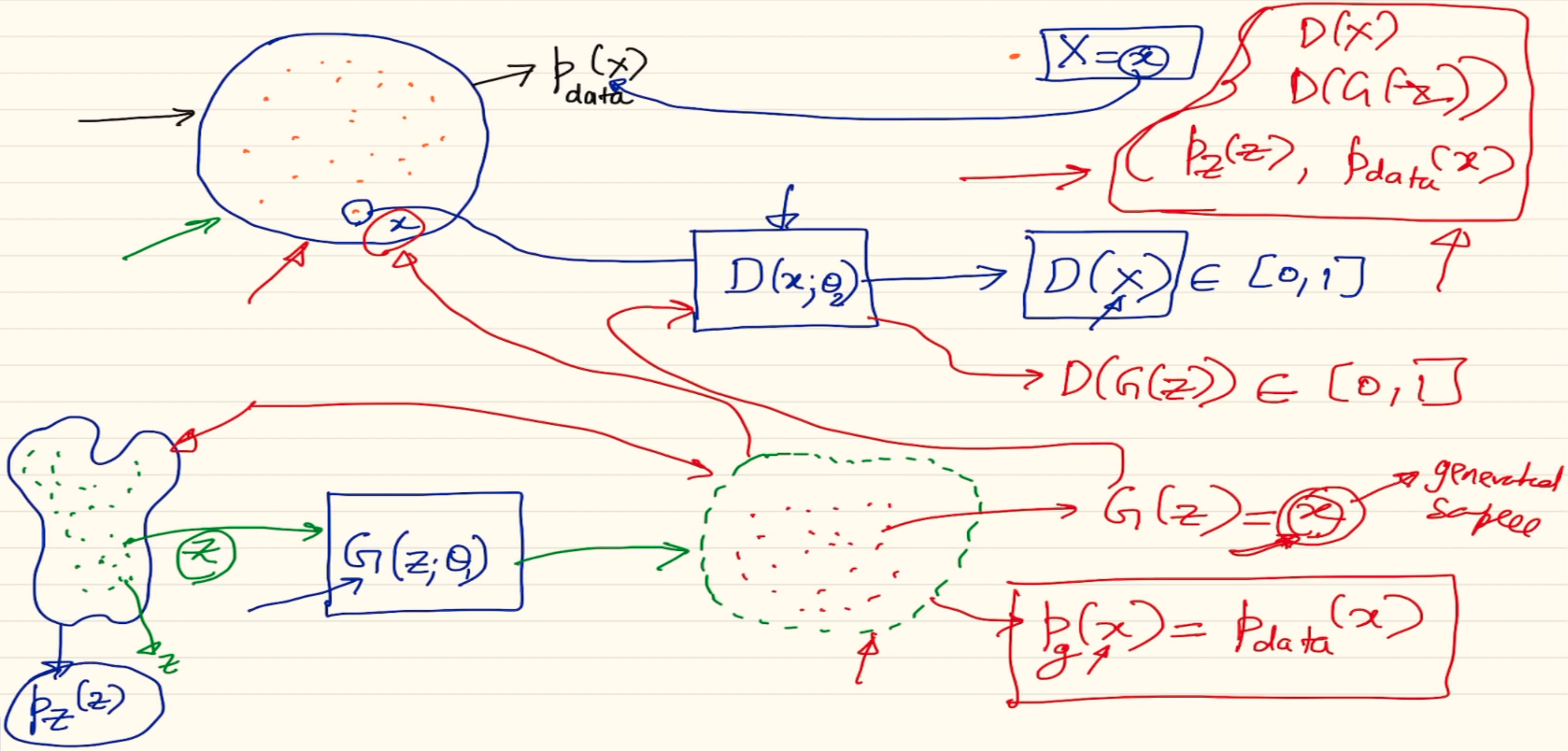
Ahlad Kumar



8:35 / 33:21

5:55 PM Fri Feb 1
27%
Notebook (6) application_form_IIITNR Attachments Notebook (7) Notebook (8)
Ahlad Kumar

Basic Conventions to understand Loss function



6:14 PM Fri Feb 1

Notebook (6) | application_form_IIITNR | Attachments | Notebook (7) | Notebook (8)

Ahlad Kumar

Loss function (Binary cross-entropy)

$\hat{y}$ = reconstructed image

y = original images

$$L(\hat{y}, y) = [y \log \hat{y} + (1-y)\log(1-\hat{y})]$$

The label for the data coming from $P_{data}(x)$ is $y = 1$

& $\hat{y} = D(x)$ so putting this we obtain

$$L(D(x), 1) = \log(D(x)) \quad \text{— (A)}$$

& for data coming from generator the label is $\boxed{y=0}$ & $\hat{y} = D(G(z))$

So in that case

$$L(D(G(z)), 0) = (1-0)\log(1-D(G(z))) = \log(1-D(G(z)))$$

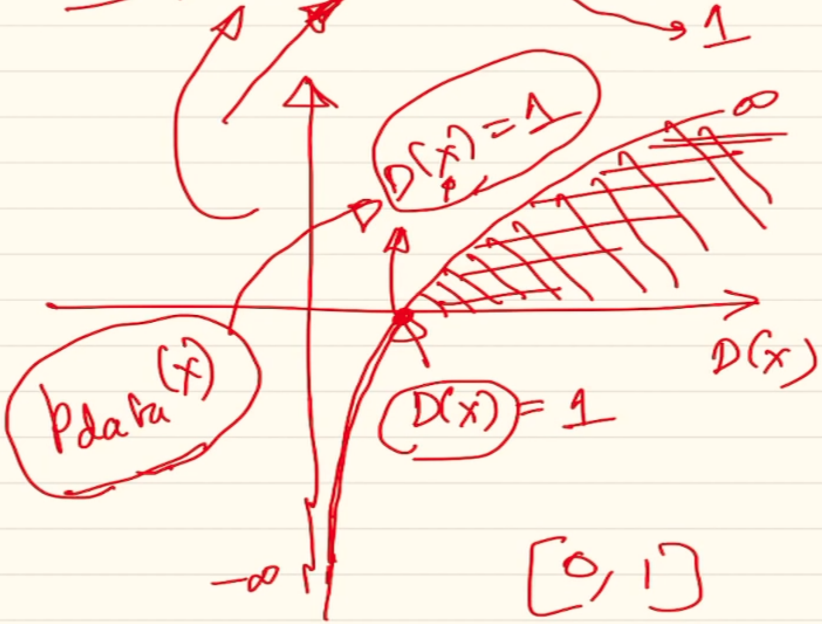
(B)

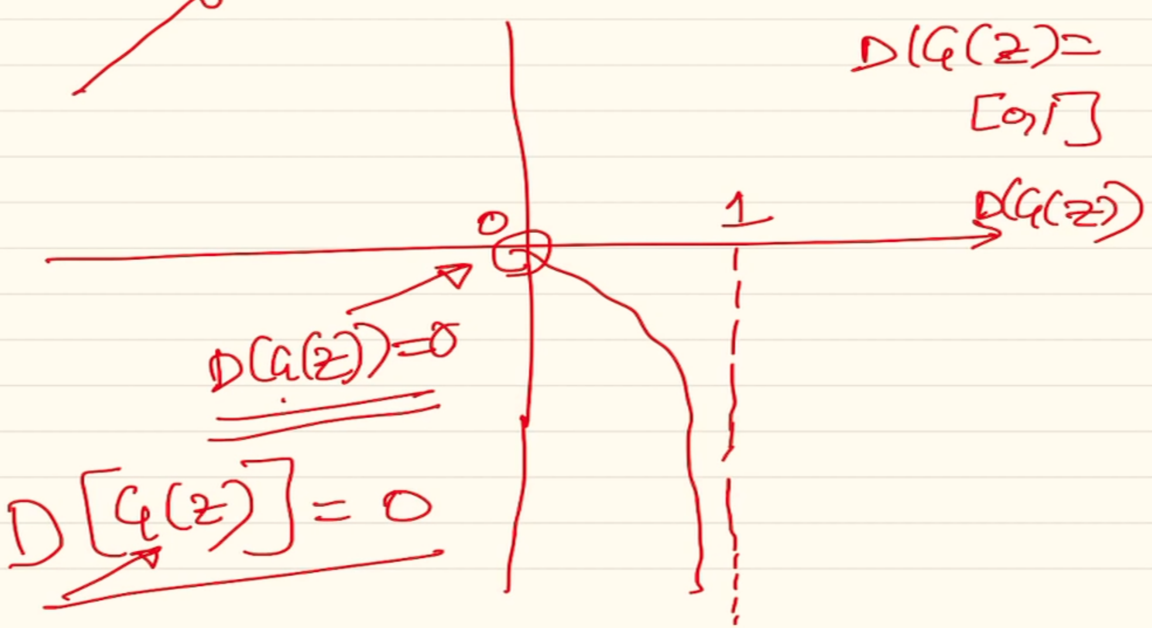
6:18 PM Fri Feb 1 22%
Notebook (6) | application_form_IIITNR | Attachments | Notebook (7) | Notebook (8)
Ahlad Kumar

Objective of the discriminator is to correctly classify fake vs the real dataset. For this (A) & (B) should be maximized

(A) $[\log(D(x))]$ → 1

$[\log(1 - D(G(z)))]$ → 0





6:36 PM Fri Feb 1

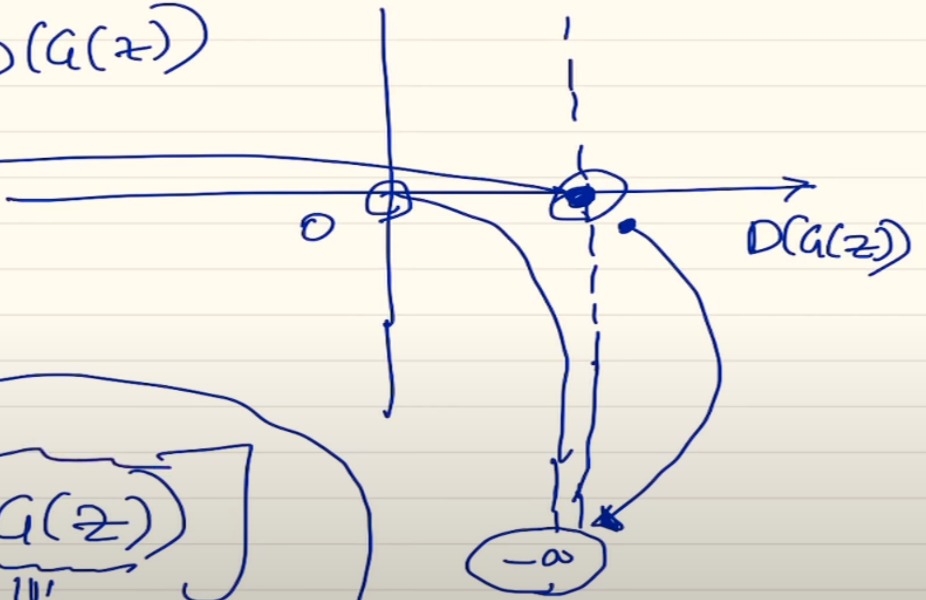
Deep Learning 28: (2) Generative Adversarial Network (GAN) : Loss Derivation from Scratch

Ahlad Kumar

$$\boxed{\max \left\{ \log(D(x)) + \log(1 - D(G(z))) \right\}} = D$$

$$D(G(z)) = 1 \rightarrow \log(1 - D(G(z)))$$

$$D(G(z)) = 1$$



$$\min \left[\log(D(x)) + \log(1 - \underbrace{D(G(z))}_{=1}) \right]$$

26:44 / 33:21

6:41 PM Fri Feb 1

Deep Learning 28: (2) Generative Adversarial Network (GAN) : Loss Derivation from Scratch

Notebook (6) | application_form_IIITNR | Attachments | Notebook (7) | Notebook (8)

Ahlad Kumar

$$\min_{G} \max_{D} \left\{ \log(D(x)) + \log(1 - D(G(z))) \right\}$$

Goodfellow 2014.

$$\min_{G} \max_{D} V(D,G) = \mathop{E}_{x \sim p_{data}(x)} [\log(D(x))] + \mathop{E}_{z \sim p_z(z)} [\log(1 - D(G(z)))]$$

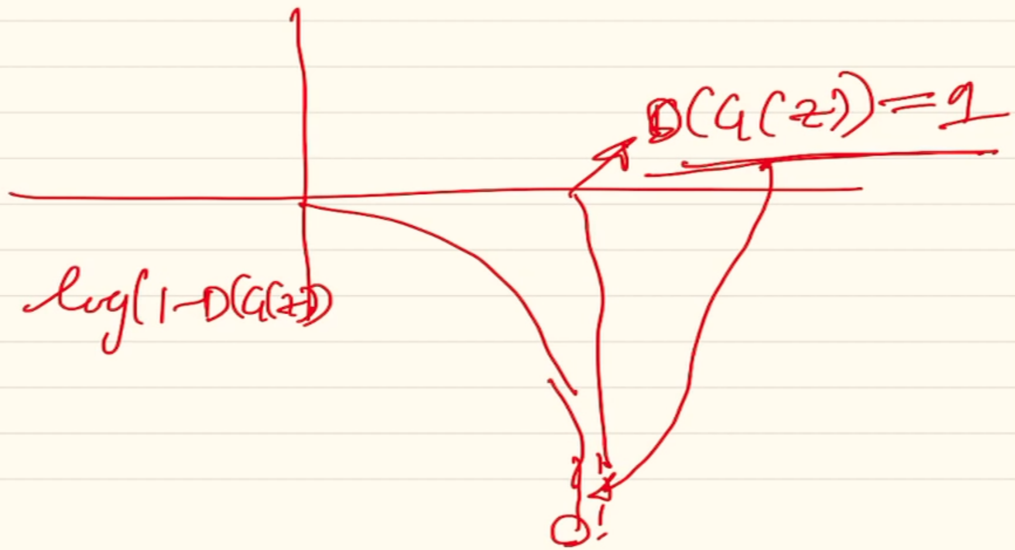
29:09 / 33:21

6:52 PM Fri Feb 1
Notebook (6) | application_form_IIITNR | Attachments | Notebook (7) | Notebook (8)
Ahlad Kumar

$$\boxed{y=0}\;,\quad \hat{y} = \boxed{D(G(z))}$$

$$L(\underbrace{D(G(z))}_{\hat{y}}, 0) = \min_G (1)\log\left[(1-D(G(z)))\right]$$

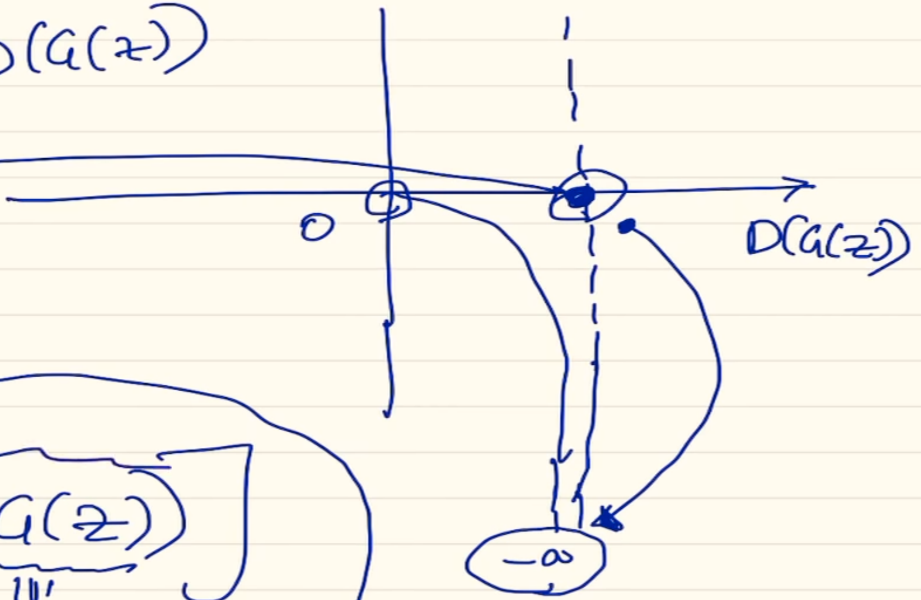
$$D[G(z)] = 1$$



Ahlad Kumar

$$\boxed{\max \left\{ \log(D(x)) + \log(1 - D(G(z))) \right\}} = D$$

$$D(G(z)) = 1 \longrightarrow \log(1 - D(G(z)))$$



$$D(G(z)) = 1$$

$$\min \left[\log(D(x)) + \log(1 - \underbrace{D(G(z))}_{=1}) \right]$$

Ahlad Kumar

$$\min_{G} \max_{D} \left\{ \log(D(x)) + \log(1 - D(G(z))) \right\}$$

Goodfellow 2014.

$$\min_{G} \max_{D} V(D,G) = \left\{ E_{x \sim p_{data}(x)} [\log(D(x))] + E_{z \sim p_z(z)} [\log(1 - D(G(z)))] \right\}$$