Author : Saurabh , 22_Jan.

Conditinal Generative Adversarial Networks

CGANs are allowed to generate images that have certain conditions or attributes.

Architecture:

- Generator : (An artist) neural network
- Discriminator : An art critic) neural network

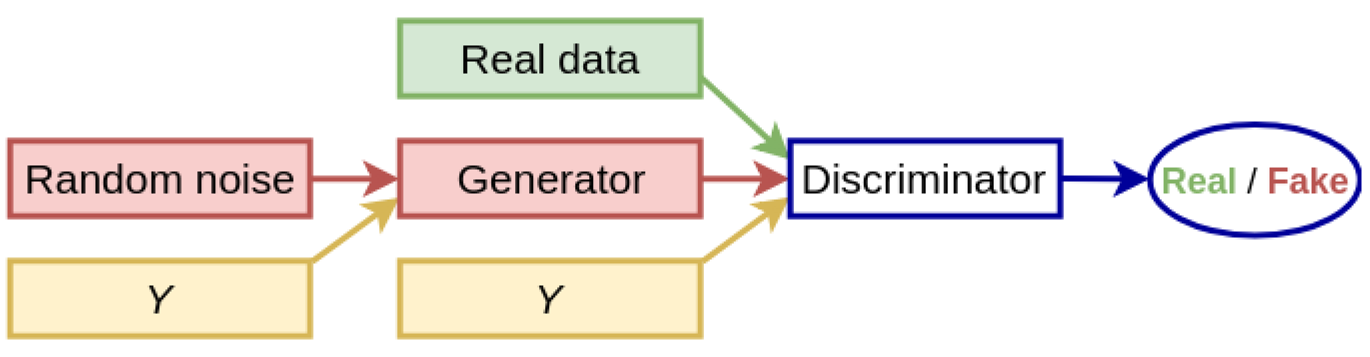
Conditional GANs (CGANs): The Generator and Discriminator both receive some additional conditioning input information. This could be the class of the current image or some other property.

Example, if we train a DCGANs to generate new MNIST images, There is no control over which specific digits will be produced by the Generator. There is no mechanism for how to request a particular digit from the Generator. This problem can be addressed by a variation of GAN called Conditional GAN (CGAN). we could add an additional input layer with values of one-hot-encoded image labels.

- Adding a vector of features controls the output and guide Generator figure out what to do.
- Such a vector of features should derive from a image which encode the class(like an image of a woman or a man if we are trying to create faces of imaginary actors) or a set of specific characteristics we expect from the image (in case of imaginary actors, it could be the type of hair, eyes or complexion).
- We can incorporate the information into the images that will be learned and also into the Z input, which is not completely random anymore.
- Discriminator's evaluation is done not only on the similarity between fake data and original data but also on the correspondence of the fake data image to its input label (or features)
- We can use the same DCGANs and imposed a condition on both Generator's and Discriminator's inputs. The condition should be in the form of a one-hot vector version of the digit. This is associated with the image to Generator or Discriminator as real or fake.

CGANs have one disadvantage. CGANs are not strictly unsupervised and we need some kind of labels for them to work.

High-Level CGAN's Architecture Diagram

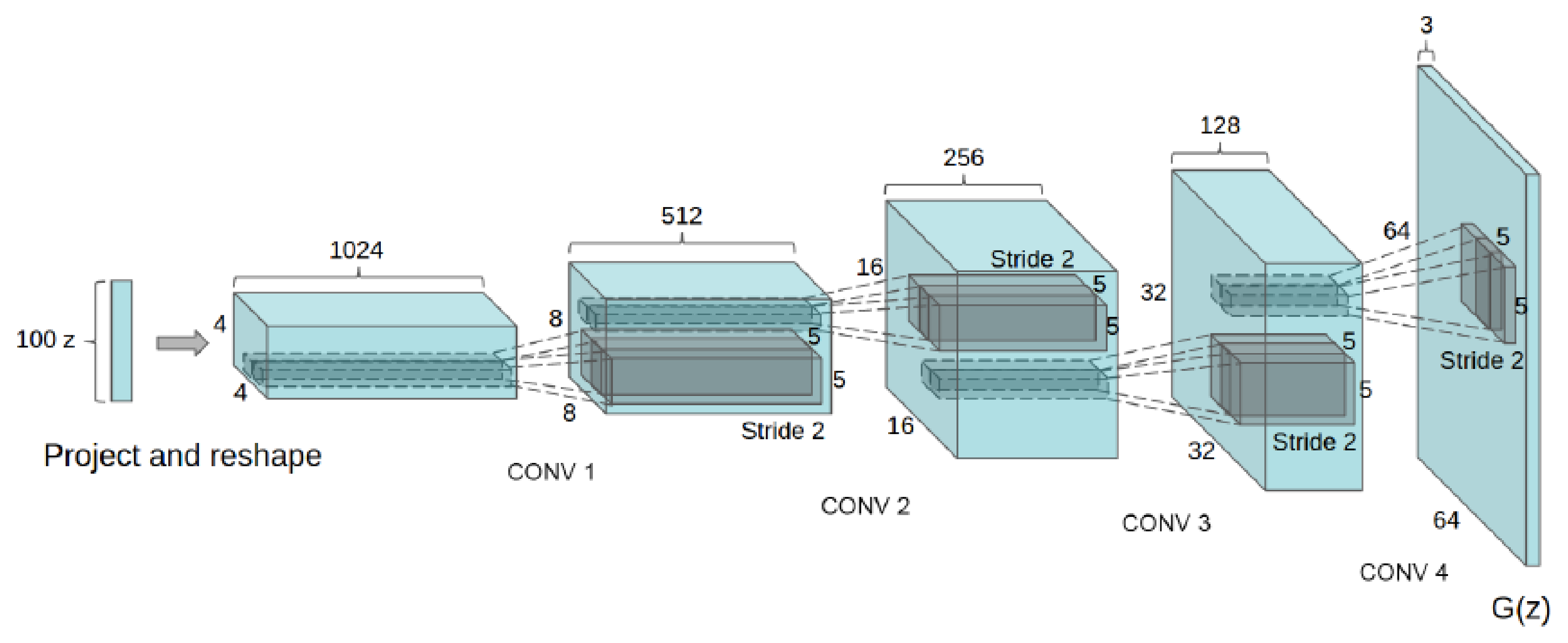


The Discriminator's Network

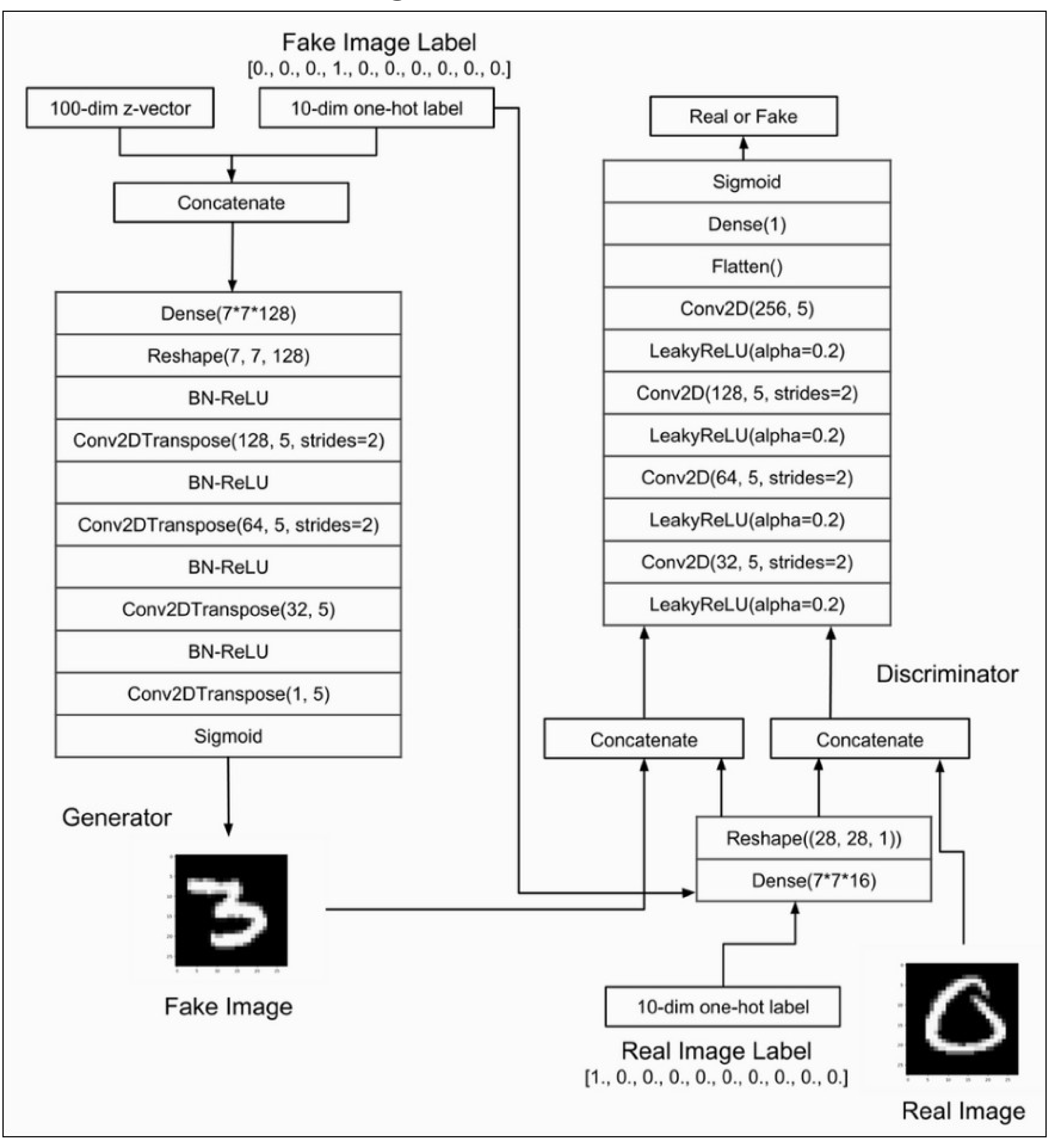
The CGAN Discriminator's model is similar to DCGAN Discriminator's model except for the one-hot vector, which is used to condition Discriminator outputs.

The Generator's Network

The CGAN Generator's model is similar to DCGAN Generator's model except for the one-hot vector, which is used to condition Generator outputs.



CGAN's Architecture Diagram



Loss Functions

The Discriminator has two task:

- Discriminator has to correctly label real images which are coming from training data set as "real".
- Discriminator has to correctly label generated images which are coming from Generator as "fake".

Discriminator The sum of the "fake" image and "real" image loss is the overall Discriminator loss. So the loss function of the Discriminator is aiming at minimizing the error of predicting real images coming from the dataset and fake images coming from the Generator given their one-hot labels.

Loss Function for Discriminator :

$$\mathcal{L}^{(D)}\left(\theta^{(G)},\theta^{(D)}\right)=-\mathbb{E}_{x\sim p_{data}}\log\mathcal{D}\left(x\mid y\right)-\mathbb{E}_{z}\log\left(1-\mathcal{D}\left(\mathcal{G}\left(z\mid y'\right)\right)\right)$$

The Generator network has one task

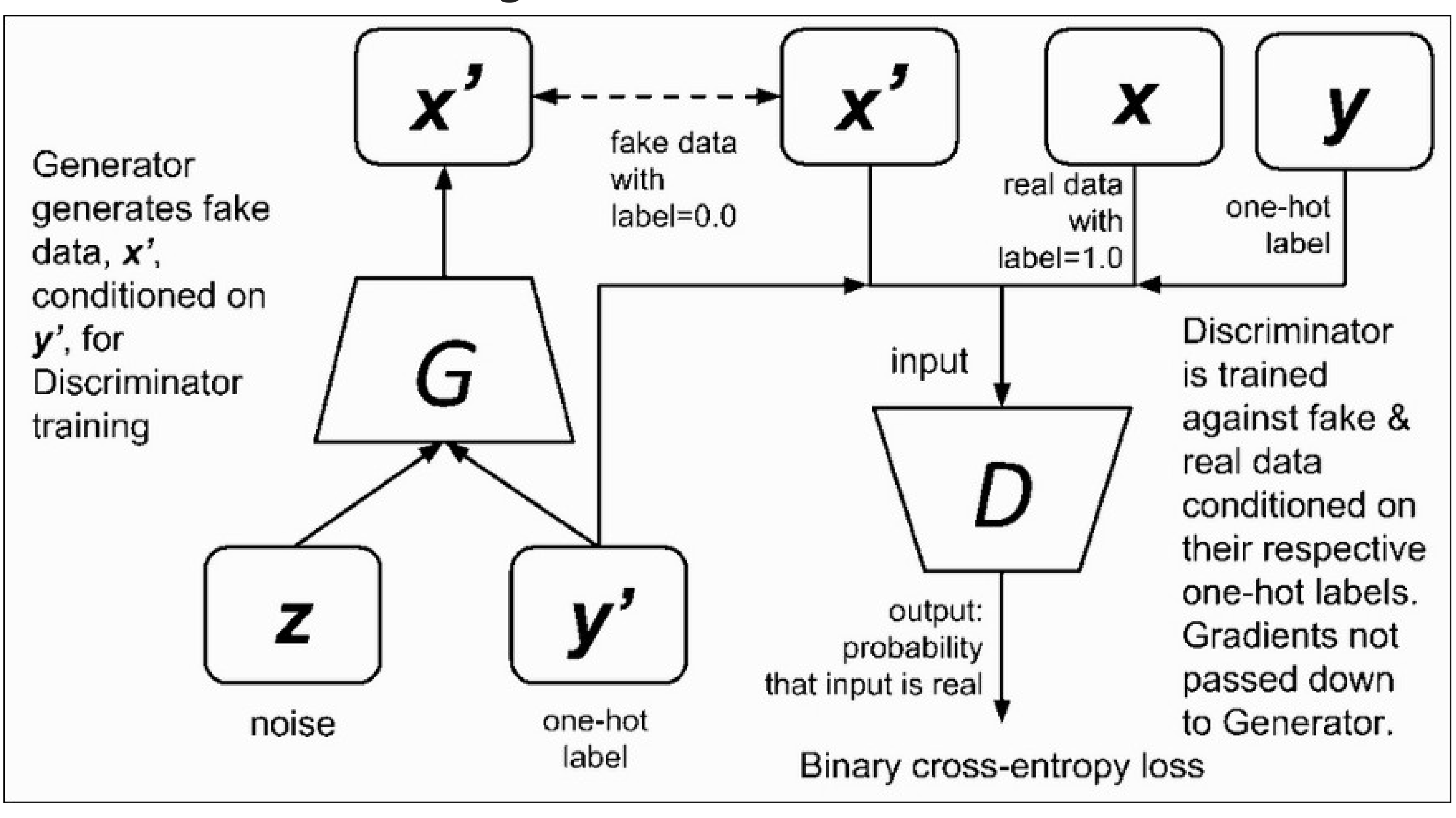
- To create an image that looks as "real" as possible to fool the Discriminator.

The loss function of the **Generator** minimizes the correct prediction of the Discriminator on fake images conditioned on the specified one-hot labels.

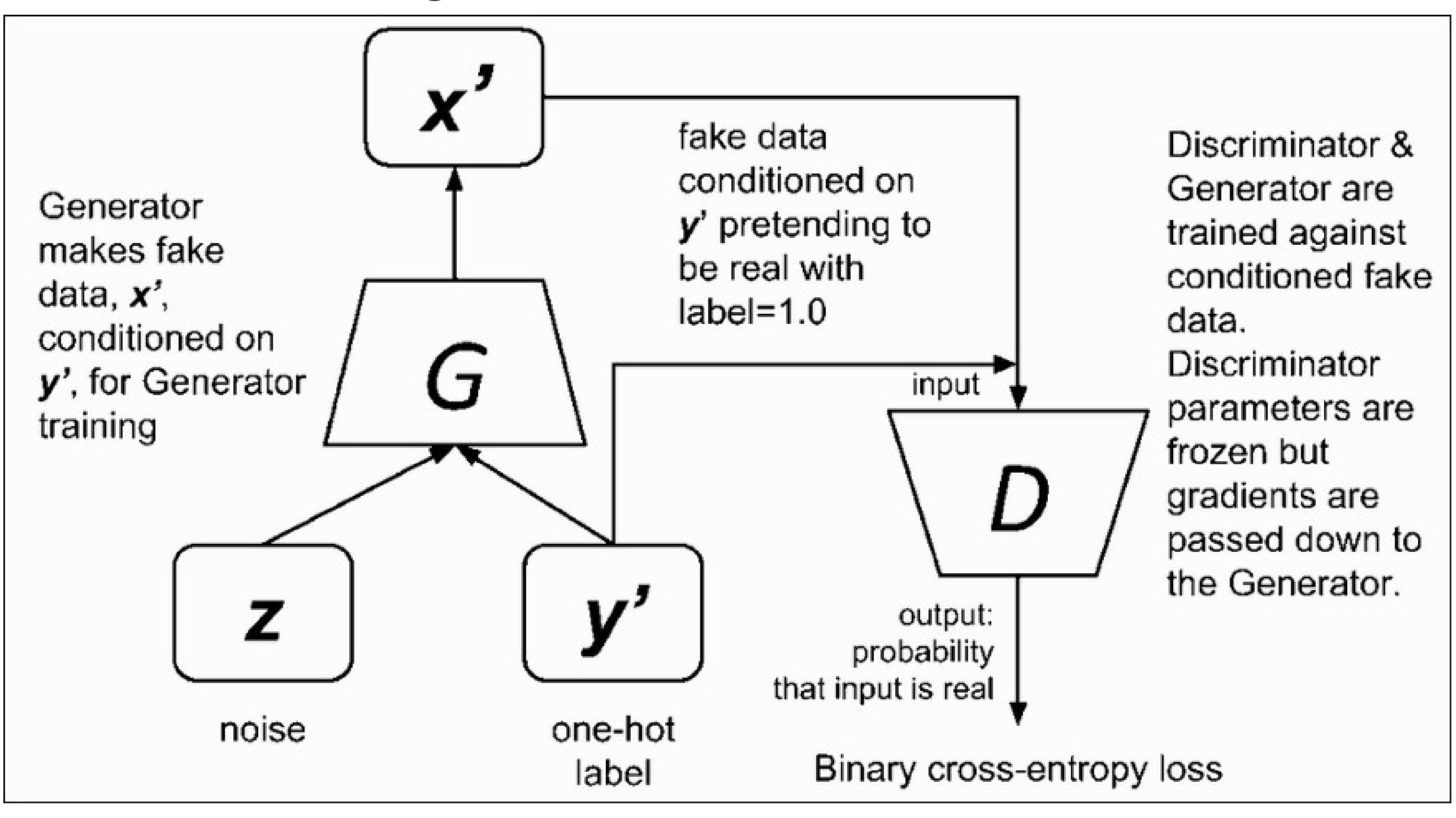
Loss Function for Generator :

$$\mathcal{L}^{(G)}\left(\theta^{(G)},\theta^{(D)}\right)=-\mathbb{E}_{z}\log D\left(\mathcal{G}\left(z\mid y'\right)\right)$$

Discriminator's Training Flow



Generator's Training Flow



Conclusion

CGANs can be used to build a model which can generate an image of an imaginary actor of given class like male or female. It can also use to build Face Aging system, Age synthesis and age progression have many practical industrial and consumer applications like cross-age face recognition, finding lost children, entertainment, visual effects in movies.

!!! Thanks for Reading

In [ ]: