

Nirbhay Sharma (B19CSE114)

dl-quiz-4

1. yes, I agree with the statement that gan can solve inference queries like $p(x)$ and $p(z/x)$, the first one looks more related to normal dcgan output and second one is more focused on given x (sample) can i generate the output from z that is belonging to x in other words we are kind of guiding it to generate from random z given a sample, so we want it to generate images belonging to that sample x and gan's can easily solve these inference queries, also these two means two different distributions $p(x)$ and $p(z/x)$ and gan's can easily learn those distributions and hence inference can be done on these.
2. yes, I agree that, dropout should not be used in Generator because if we see the conceptual detail of dropout it makes weight values zero for some neurons with a probability p so only some of the neurons learn at a particular time, so if we apply dropout, then the generator will not be able to generate images of good quality and also will not be able to learn the distribution well enough to beat the discriminator and hence dropout in generator is not recommended.
3. the GAN architectue that I liked the most is pix2pix GAN or I2I translation gan, the working of this gan is as follows
 - the I2I gan is a version of conditional gan in which we are guiding generator to what to generate
 - In I2I gan the discriminator architecture is same as traditional (a classifier of real and fake)
 - the architecture of generator changes, in generator we will have a full encoder-decoder model where the image is passed to the encoder and from it's latent representation generated by encoder it generates the image using the decoder
 - once the image is generated from the generator then, it needs to pass to the discriminator with another image which is the ground truth image for that task, if we take an example of image coloration then the ground truth, colored image and the generator generated color image is passed to the discriminator for real/fake classification
 - one major thing also is to modify the loss function which will now become (adversarial loss + L_1 loss)
 - apart from the traditional GAN loss ($\log(D(x)) + \log(1 - D(G(x)))$) we are adding L_1 loss because we also need the generator to generate images that are similar to ground truth images and hence to train the generator also we are adding this l1 loss
 - in this manner the pix2pix gan is trained with slight modifications in generator, loss function
4. the hyperparameters for gan includes
 - learning rate for generator and discriminator
 - batch_size
 - parameters for optimizers (say β_1, β_2)
 - random noise to the generator can also be considered as a hyper parameter
 - loss function

- initial weights for generator and discriminator
- epochs

one more thing to say here is that Gan's are generally more sensitive to hyperparameters and all these hyperparamters tuning together made the gan to converge.
