Generating Images with a Conditional Generative Adversarial Network

Anonymous author

Sidharth Rao]

Abstract

In this study, conditional GANs and generative adversarial networks (GANs) are introduced, with a particular emphasis on how deep learning can be used to generate images. Using a generator network that has been trained to create samples that are realistic enough to trick a discriminator network, GANs are a form of neural network architecture that can learn to generate fresh data. On the other hand, conditional GANs add a conditioning variable that enables focused production of particular image kinds. The main characteristics of GANs and Conditional GANs are examined, along with their advantages and disadvantages in picture generating tasks. Also it focuses at some of the current developments and difficulties in this field of study.

1 Methodology

They are made up of two "adversarial" models: a discriminative model D that calculates the likelihood that a sample came from the training set rather than the generative model G, which represents the distribution of the data. Both G and D might be a multi-layer perceptron or another non-linear mapping function. The generator constructs a mapping function from a prior noise distribution pz(z) to data space as G(z; g) in order to learn a generator distribution pg over data data x. And the discriminator, D(x; d), produces a single scalar that indicates the likelihood that x originated from training data as opposed to pg. [1]

$$\min_G \max_D V(D,G) = \mathbb{E}x \sim pdata(x)[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))] \tag{1}$$

[1]

If both the generator and discriminator are dependent on some additional information, y, such as class labels or input from other modalities, generative adversarial nets can be expanded to a conditional model. By adding y as an additional input layer to the discriminator and generator, we can condition the system. 2 The adversarial training framework provides for a great deal of freedom in how this hidden representation is put together. In the generator, the prior input noise, pz(z), and y are merged to form a joint hidden representation. 1 In the discriminator, x and y are given as inputs to a discriminator function. [1]

The objective function of a two-player minimax game would be as Eq 2

$\min_G \max_D V(D,G) = \mathbb{E}x \sim pdata(x|y)[\log D(x|y)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z|y)))]$. [1]

Figure 1 shows how the GAN is trained in 1 cycle:

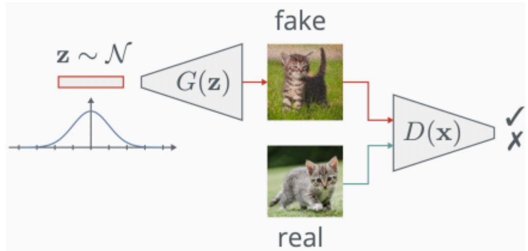
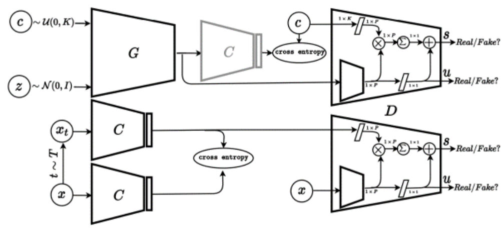


figure 1 is from this lecture: https://cwkx.github.io/materials/deep-learning/dl-lecture4.pdf

Here is another diagram which illustrates how the model works:



[2]

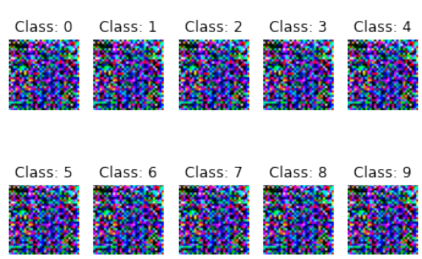
The model is trained using the train() function, which takes the number of epochs as input. In each epoch, the function loops over the batches of the training data, trains the discriminator using both real and fake images, and trains the generator by maximizing the discriminator's probability of the generated images being real. The generator is trained to fool the discriminator by generating realistic images that are difficult to distinguish from the real ones.

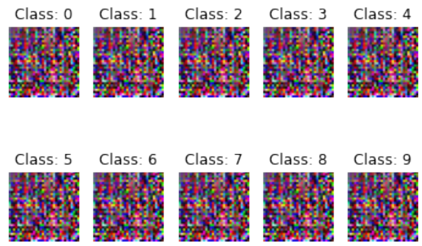
The model uses various deep learning techniques, including convolutional neural networks (CNNs), batch normalization, leaky ReLU activation functions, and dropout regularization, among others. The generator uses a series of transpose convolutional layers to generate the images, while the discriminator uses a series of convolutional layers to classify them.

figure 1 is from this lecture: https://cwkx.github.io/materials/deep-learning/dl-lecture4.pdf

2 Results

These are 2 different epochs of image generation from the GAN. Due to the limitations of run time and lack of hyperparamater tuning, there was no improvement in the accuracy of the images generated.





unfortunately there were no interpolations to display and due to the limited training time there were no "best" images produced.

3 Limitations

The results are inconsistent and gargled. In the experiment, due to time constraints, the model was unable to be trained for a sufficient number of epochs to achieve optimal performance. As a result, the generated images produced by the model were not of the desired quality, and the overall performance of the model was below expectations. It is important to note that future studies may need to take this into consideration and allocate more time and resources for training the model to obtain better results.

Bonuses

This submission has a total bonus of -12 marks (a penalty), as it is trained only on CIFAR-10 using a GAN model.

References

[1] Diederik P Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". In: *arXiv preprint arXiv:1411.1784* (2014).

[2] Steven Liu et al. "Diverse image generation via self-conditioned gans". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 14286–14295.