CGAN

A Preprint

Abstract

1 Architecture

CGAN consists of both Generator(G) and Discriminator(D) as in default GAN, however it has a signifact change in terms of adding condition.

Generator: Input Combination: The noise vector z and the conditioning information y are combined into a single, larger input vector. The most straightforward way this is done, and what Figure 1 implies for the initial layers, is concatenation. So, if z has 100 dimensions and y has 10 dimensions (for 10 digit classes), they would be concatenated to form a 110-dimensional input vector.Neural Network Transformation: This combined vector is then fed as input into the generator's neural network (typically a Multi-Layer Perceptron or a more complex architecture like a deconvolutional network for images). Learning: The network's weights are adjusted during training to learn how to transform this combined (noise + condition) input into a data sample that resembles real data corresponding to the condition y.

Discriminator: Discriminator (D):

Goal: To distinguish between real data samples (from the training set) and fake data samples (produced by the generator), given the conditioning variable y. It needs to determine if a sample is authentic for that specific condition. Inputs: x: A data sample. This can be a real sample from the training data or a fake sample G(z|y) from the generator. y: The conditioning variable associated with the sample x. If x is real, y is its true label/condition. If x is fake, y is the condition used by the generator to create it. Process (with combined inputs): Input Combination: The data sample x (e.g., a 784-pixel MNIST image) and the conditioning information y (e.g., the one-hot vector for "7") are combined into a single input representation. Again, concatenation is a common and implied method here. Neural Network Transformation: This combined vector is fed as input into the discriminator's neural network. Learning: The network's weights are adjusted during training to learn how to output a high probability if the input sample x is real and matches the condition y, and a low probability if x is fake, given y. Output: D(x|y): A single scalar value (a probability) representing the likelihood that the input sample x is a real data sample from the true data distribution, given the condition y.

2 Details

The conditioning variable y is not just an afterthought; it's integrated directly with the primary input (noise z for G, data x for D) at the input stage of their respective networks. This allows the networks to learn mappings that are explicitly dependent on y, enabling the controlled, conditional generation and discrimination that is the hallmark of cGANs. The diagram (Figure 1) clearly shows z and y feeding into G(z|y), and x and y feeding into D(x|y), visually representing this joint processing from the very first step.