Teaching Notes: DCGAN

Prepared by: Abhijit Singh Jowhari

Introduction

These notes provide an overview of the Deep Convolutional Generative Adversarial Network (DCGAN), focusing on its architecture, the roles of the generator and discriminator, and the training process. DCGAN is a type of GAN that uses deep convolutional networks to generate high-quality images.

1 DCGAN Architecture

1.1 Generator

The generator in a DCGAN is designed to create realistic images from random noise. It uses a series of transposed convolutional layers to upsample the noise vector into a full-sized image.

1.1.1 Key Features

- **Input**: A random noise vector, typically sampled from a uniform or normal distribution. - **Layers**:

- **Transposed Convolutions**: Increase the spatial dimensions of the data.
- **Batch Normalization**: Stabilizes and speeds up the training process.
- **Activation Functions**: ReLU activations are used in all layers except the last, where a Tanh activation is applied.

1.1.2 Example Architecture

- Input: 100-dimensional noise vector z
- Layer 1: Transposed Convolution + Batch Norm + ReLU
- Layer 2: Transposed Convolution + Batch Norm + ReLU
- Layer 3: Transposed Convolution + Batch Norm + ReLU
- Output Layer: Transposed Convolution + Tanh

1.2 Discriminator

The discriminator in a DCGAN is a binary classifier designed to distinguish between real and fake images. It uses a series of convolutional layers to down-sample the input image and extract features, followed by a fully connected layer to output a probability score.

1.2.1 Key Features

- **Input**: An image, either real or generated. - **Layers**:

- **Convolutions**: Extract features from the input image.
- **Batch Normalization**: Stabilizes training.
- **Activation Functions**: Leaky ReLU activations are commonly used.
- **Output Layer**: Fully connected layer followed by a sigmoid activation function to output a probability score between 0 (fake) and 1 (real).

1.2.2 Example Architecture

- Input: Image (e.g., 64x64x3)
- Layer 1: Convolution + Leaky ReLU
- Layer 2: Convolution + Batch Norm + Leaky ReLU
- Layer 3: Convolution + Batch Norm + Leaky ReLU
- Output Layer: Fully Connected + Sigmoid

2 Training Process

The training of a DCGAN involves an adversarial process where the generator and discriminator are trained simultaneously:

2.1 Discriminator Training

The discriminator is trained to maximize the probability of assigning the correct labels to both real and fake images. The objective is to correctly identify real images as real and fake images as fake.

2.1.1 Discriminator Loss

$$L_D = -\mathbb{E}_{x \sim p_{\text{data}}}[\log D(x)] - \mathbb{E}_{z \sim p_z}[\log(1 - D(G(z)))] \tag{1}$$

2.2 Generator Training

The generator is trained to minimize the probability of the discriminator correctly identifying fake images. The objective is to generate images that are realistic enough to fool the discriminator into classifying them as real.

2.2.1 Generator Loss

$$L_G = -\mathbb{E}_{z \sim p_z}[\log D(G(z))] \tag{2}$$

Conclusion

These notes provide an overview of the DCGAN model, highlighting its generator and discriminator architecture, as well as the adversarial training process. Understanding these components is crucial for grasping how DCGANs generate high-quality images.

References

[1] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y. (2014). Generative Adversarial Nets. In *Advances in Neural Information Processing Systems* (pp. 2672-2680).

[2] Radford, A., Metz, L., Chintala, S. (2015). Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. arXiv preprint arXiv:1511.06434.