Sangyoon Kim

CPSC393: Machine Learning

Dr. Chelsea Parlett

May 12, 2023

Differences Between Model Architectures, Inputs/Outputs, and Training

In this experiment, few generative models for images were compared based on its architecture, inputs, outputs, and training. There are several different types of models that can be used, including Variational Autoencoders (VAE), Generative Adversarial Networks (GAN), and Conditional Generative Adversarial Networks (Conditional GAN).

VAE is a type of generative model that learn a compressed representation of the input data, called the latent space. It consists of an encoder that maps the input data to the latent space, and a decoder that maps the latent space back to the original data. A VAE learns a probability distribution over the latent space. The hidden representation layer is probabilistic. In VAE, the concept vector represents a set of vectors in the latent space that correspond to features of the input data. The VAE learns to map the input data to a latent space. The latent space can be thought of as a compressed representation of the input data, where each point in the latent space corresponds to a possible encoding of the input data. The concept vectors allow to identify features or attributes of the input data that are important for the downstream, and then manually specify the values of these features in the latent space. For example, in a VAE trained on images of faces, one might define concept vectors for attributes such as gender, age, or facial expression, by specifying the corresponding values of these attributes in the latent space. Once the concept vectors have been defined, they can be used to manipulate the latent space to generate new samples with specific attribute values. For example, one can add or subtract concept vectors to the latent space representation of an image to modify its attributes. During training, the model learns to minimize the reconstruction loss between the input data and the reconstructed data, as well as the KL divergence between the learned latent distribution and the prior distribution. The model can be used to sample points from the latent space and then generate data by passing theses points through the decoder to generate new samples. The VAE tend to produce blurry images and struggle with generating highly realistic images, but they are good at interpolating between different samples. The ideal latent space should have three properties including representation, realism, and smoothness. First, the latent space should be able to represent the important underlying factors of variation in the input data. Ideally, each dimension of the latent space would correspond to a single feature of the data. It enables the VAE to manipulate individual features of the input data by changing specific dimensions in the latent space. Next, the VAE model should be able to generate realistic samples from the latent space. The generated samples should be visually similar to the input data and should exhibit a high degree of variation. Lastly, the latent space should be smooth and continuous, meaning that small changes

in the latent vector should result in gradually changes in the generated output. This enables the VAE to interpolate between different samples in the latent space, allowing it to generate new samples that are intermediate between two existing samples. Smoothness is also important for enabling gradient-based optimization methods to work effectively in the latent space.

GAN is a type of generative model that consist of two neural networks including a generator that generates new samples, and a discriminator that tries to distinguish between real and generated fake samples. These two are trained together in a process called adversarial training, where the generator tries to produce samples that fool the discriminator, while the discriminator tries to correctly classify real and fake samples. The non-saturating GAN loss is a common loss function used in the model that addresses the problem of vanishing gradients during training. This loss function is commonly used with the generator, while the discriminator uses a binary cross-entropy loss function. In the non-saturating GAN loss, the objective is to maximize the probability that the discriminator outputs a value of 1 for the generated data, while the generator tries to generate data that the discriminator will classify as 1. The loss function is defined as $L = -log(D(G(z))$. In the function, $D(G(z))$ represents the discriminator's output for the generated data G(z), and z represents a random noise vector used as input to the generator. The loss function encourages the generator to produce data that is classified as real by the discriminator. Therefore, it allows to avoid the problem of vanishing gradients that can occur when using other loss functions such as the original GAN loss. The GAN can produce high-quality, realistic images, but they can be difficult to train. One of the common challenges in training the model is the lack of convergence. It can occur for several reasons. The mode collapse, where the generator only produces a limited set of outputs. It can happen when the discriminator is too strong and can easily identify the generator's output as fake. As a result, the generator may learn to produce only a few outputs that can fool the discriminator. Alternative loss functions such as the Wasserstein loss can help stabilize training and prevent mode collapse. GAN is good at generating diverse samples, but they can struggle with interpolating between different samples.

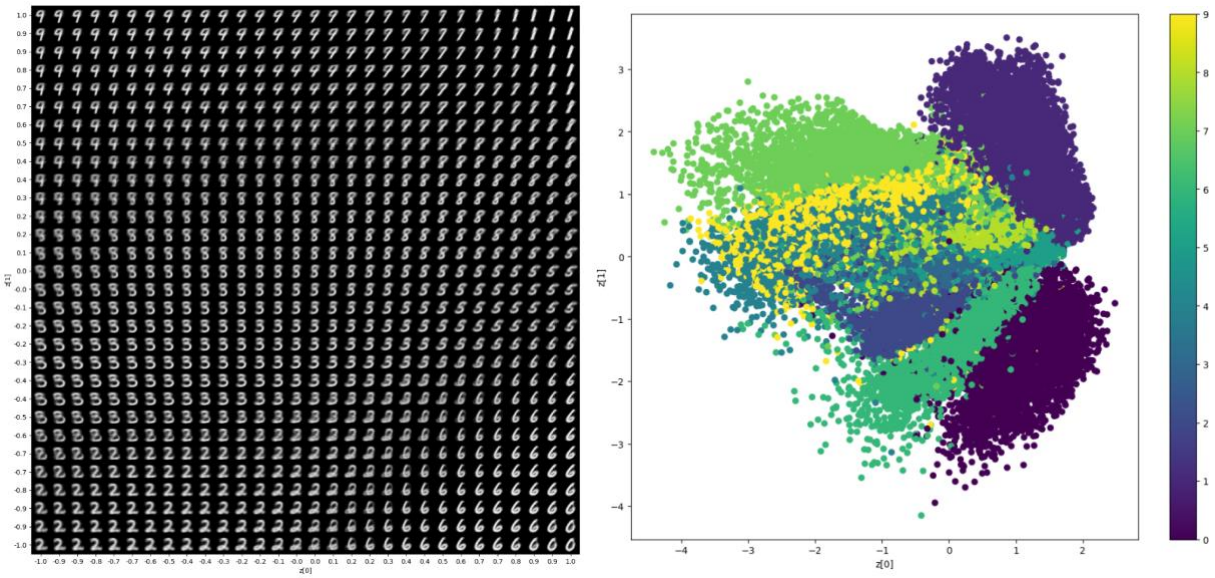
Conditional GAN is a type of GAN that can generate images conditioned on some additional input information, such as class labels, text descriptions, or images. Conditional GAN consists of a generator and a discriminator, similar to the traditional GAN model. However, the main difference between GAN and conditional GAN is the way they generate data. In addition to the generator taking a random noise vector as input, it also takes in some additional information that is used to condition the output. In other words, GAN generate data without any constraints or control, while conditional GAN generate data conditioned on some additional input information. For example, in a Conditional GAN that generates images of different animals, the conditioning input might be a one-hot vector indicating the type of animal to be generated. Also, the conditional information in conditional GAN can be used to control the generated data and make it more specific to a certain class or attribute. For example, in an image generation task, the conditional information can be the label of the object in the image, such as "cat" or "dog". This enables the generator to generate images that are specific to the given label. During training, the generator tries

to produce images that are not only realistic but also match the conditioning input, while the discriminator tries to distinguish between real and generated images, taking into account the conditioning input. Conditional GAN can produce good quality, diverse images that match the conditioning input. They are also good at interpolating between different conditioning inputs to generate intermediate images. However, Conditional GAN can be more difficult to train than traditional GAN, especially when the conditioning input is complex or high-dimensional. Additionally, the quality of the generated images can be highly dependent on the quality and relevance of the conditioning input.

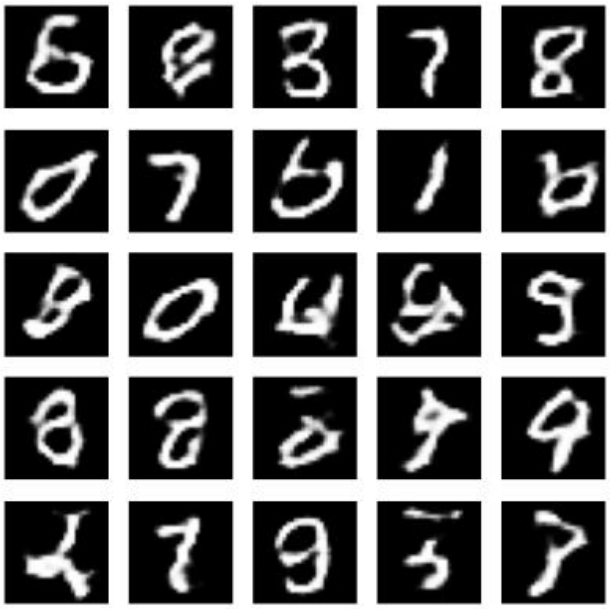
Regard to their strengths and weaknesses, VAE is particularly good at learning the underlying distribution of the data and generating realistic samples from the learned distribution. However, they may generate blurry or less realistic samples compared to GAN or Conditional GAN. GAN and Conditional GAN, on the other hand, are particularly good at generating highly realistic samples, but they may suffer from instability during training and may not be as good at capturing the underlying distribution of the data as VAE.

Comparison of Outputs

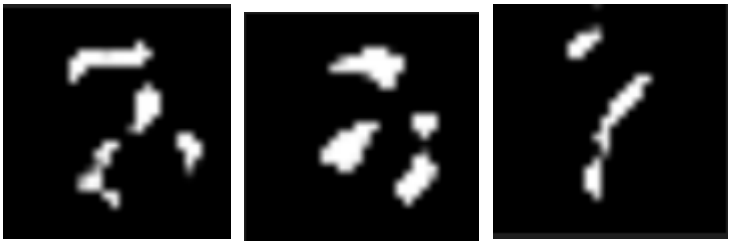
Output from VAE:



Output from GAN:



Output from Conditional GAN:



VAE generates output by encoding input data into a low-dimensional latent space and then decoding it back into the original data space. The output from VAE can be similar to the input data, but with some degree of variation and randomness due to the nature of the latent space. The output from VAE was particularly good at generating new samples that are similar to the training data, but with some variation. The graph suggests that the model achieved a balance between faithful reconstruction and meaningful latent representations by using a using a combination of the reconstruction loss and KL divergence.

GAN generates output by training a generator to produce fake data that can fool a discriminator. The output from GAN was highly realistic and even almost be indistinguishable from real data. However, because GAN does not have a deterministic mapping from the input noise vector to the output data, the generated samples could somewhat highly variable even for a fixed input noise vector. In this model, the generator could learn to generate samples that are highly specific to the training data, which can result in overfitting.

Conditional GAN generates output that is conditioned on some additional input information, in the form of a label or class information. The output from Conditional GAN can be highly specific to the given conditional information, such as generating images of a specific object or with a specific style. Conditional GAN was particularly good at generating samples that are highly specific to the given condition.

Lastly, the subjective opinions on the differences in the generated images, it really depends on the specific model and how it was trained. In quality of the outputs, the output from GAN and Conditional GAN can be highly realistic but may lack the variability and randomness of VAE. VAE may generate more varied output but may not be as highly realistic as GAN or Conditional GAN.