Winged Horses with a Deep Convolutional Generative Adverserial Network

Anonymous author

Abstract

This paper proposes using a Deep Convolutional Generative Adverserial Network (DCGAN) to generate images that look like winged horses. The model is trained on both the CIFAR10 and STL10 datasets. It follows the same architectural guidelines provided in the original DCGAN paper, as well as experimenting with the addition of dropout layers, which appeared to decrease the appearance of the results. After a large amount of epochs, it is evident that the model was able to recreate images that appear to be from the real data distribution, as well as images containing features which span over multiple classes (winged horses).

1 Methodology

The method entails training a DCGAN [3] to generate images of winged horses, using only images of horses and birds contained within the CIFAR10 and STL10 datasets. We exclusively select only images or horses and birds to try to mitigate mode collapse. Our DCGAN defines both a deconvolutional generator G and a convolutional discriminator D, such that G and D participate in a two-player minimax game with value function $V(G, D)$ [1]:

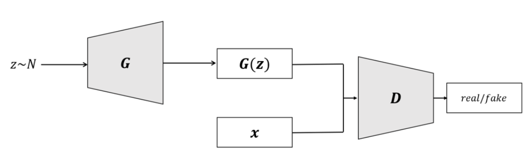
$$\min_G \max_D V(D,G) = \mathbb{E}_{x \sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))]. \tag{1}$$

Whilst this equation captures the minimax game played by the generator and discriminator, in practice, rather than minimising $\log(1 - D(G(z)))$ for the generator G, we maximise $\log D(G(z))$ instead. This provides more adequate gradients for G, ensuring our discriminator does not overpower G and simply reject its fake images with confidence [1] in the early stages of training, causing saturation of G.

By training our generator and discriminator using the minibatch stochastic gradient descent training algorithm proposed in the original GAN paper [1], our generator can map a z-dimensional noise vector (selected from a normal distribution) to a new image that contains features similar to some actual data, x.

Overtime, our generator attempts to maximise the probability that the generated images belong to the real data distribution, p_{data} providing us with better images of horses and birds (or a combination of the two).

The diagram below depicts a general overview of the DCGAN architecture:



Both the generator G and discriminator D follow the same architectural guidelines stated in the original DCGAN paper [3]. Dropout layers were experimented with in the discriminator to ultimately try and reduce mode collapse and increase train stability [2] however this came with no success.

As a winged horse constitutes of features mainly belonging to a horse, the DCGAN is trained with a higher proportion of horse images (3:1 ratio of horse to bird) to allow the generator model to better generalise the horse class. This is turn, allows generated images to better resemble horses, whilst still possibly containing features belonging to birds, i.e. beaks, wings.

2 Results

Different resolutions of our combined dataset were tested, leading to two slightly different model architectures - one that handled 32x32 images and the other 64x64. Whilst STL10 is 64x64, it was first down-sampled to 32x32 to match CIFAR10's resolution. The two images below show our results from 32x32 and 64x64 architectures after 30 epochs:

Figure 1: 32x32 birds and horses

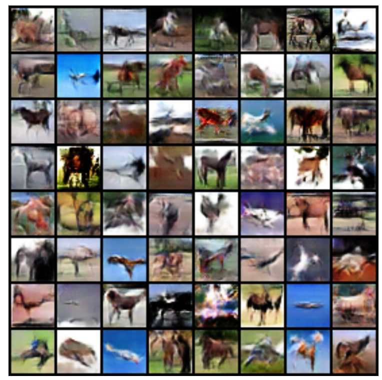


Figure 2: 32x32 birds and airplanes

The use of airplanes was discontinued due to the comparatively poor results achieved whilst using them. The figures below show the results for the 64x64 model, when training over 30 epochs and 100 epochs respectively.

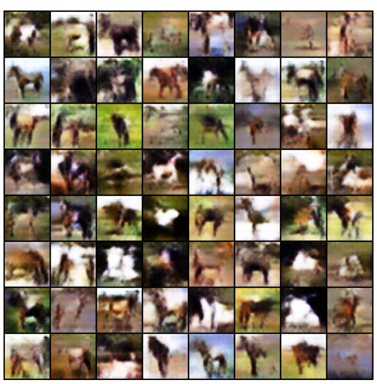


Figure 3: 64x64 birds and horses

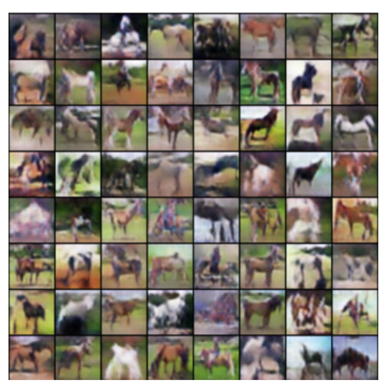
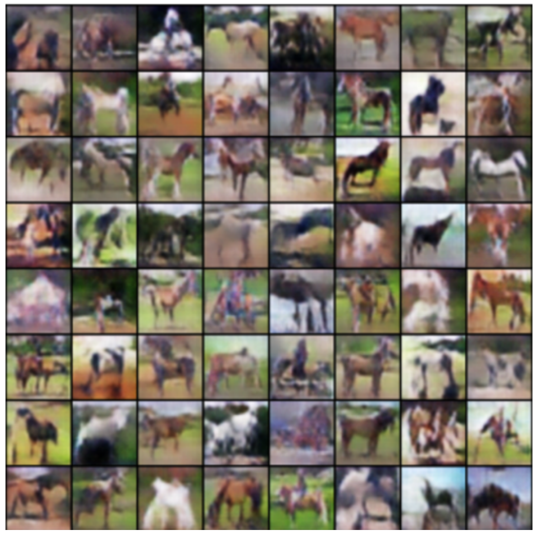
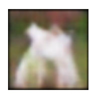


Figure 4: 64x64 birds and horses

The batch of 64 images below indicate the best results from what the generator was able to produce after 100 epochs. As expected, our best batch mainly constitutes of images of horses, due to feeding the discriminator larger proportions of horse images. There is also evidence that our generator picked up some bird-like features, as in some of the images we can identify what appears to be wings and beaks (with some stretch of the imagination).



From this batch, the most Pegasus-like image is:



3 Limitations

Both the CIFAR10 and STL10 datasets are relatively small, only containing 5000 and 500 training images per class respectively. We could expand both of these datasets by defining a classifer model that can identify horses (maybe even just white horses) and birds in the test sets to then be used by our DCGAN in addition to the test data provided.

With both CIFAR10 and STL10 providing images at different resolutions, if we want to use both, we must decide on a fixed resolution to use for both datasets. Had more higher resolution data been available, we could of trained solely on this higher resolution data and consequently produced higher quality images.

Many of the birds contained in these two datasets often have their wings closed, or are not in flight. This makes it difficult for the DCGAN to pick up on this feature. Airplanes could be substituted from birds, however, as my results show, due to the drastic differences in horse and airplane images, the DCGAN struggled to generalise the two into an image that aligns with the real data distribution.

Bonuses

This submission has a total bonus of -4 marks (a penalty) as it uses an adverserial training method.

References

[1] Ian J. Goodfellow et al. *Generative Adversarial Networks*. 2014. arXiv: `1406.2661 [stat.ML]`.

[2] Gonçalo Mordido, Haojin Yang, and Christoph Meinel. *Dropout-GAN: Learning from a Dynamic Ensemble of Discriminators*. 2020. arXiv: `1807.11346 [cs.LG]`.

[3] Alec Radford, Luke Metz, and Soumith Chintala. *Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks*. 2016. arXiv: `1511.06434 [cs.LG]`.