Exercise 7

Arnas Šniokaitis

January 26, 2026

1 Generative and discriminative models

- In one sentence each, describe the core idea of discriminative and generative models, highlighting their difference.

  Discriminative models, using feature vectors, aim to calculate/approximate some other feature which we have the goal of achieving. Generative models, using the data we provide, try to generate new data such that it would fit in as cleanly as possible with the privided data.

- Provide an example where a discriminative model might be more appropriate than a generative model. Justify your choice based on the characteristics of the data and the task.

  If our goal is to be able to, for instance, look at images of cells, and clasify whether a certain disease is present or not we would use a discriminative model. As a generative model would create new images which may have labels but it wouldn't be applicable to determine if new images show a disease or not.

- Describe the difference between explicit and implicit density estimation and what impact it has on the generation ability of a generative model.

  For explicit density estimation we solve for the probability density(esitmation) where as for implicit estimation we do not solve for the density itself, we learn to sample from it.

2 Autoencoders and Variational Autoencoders

- Explain how an autoencoder can be used for dimensionality reduction. What aspects of the data are preserved in the reduced representation?

  Let's say we have n input features and our autoencoder has $m < n$ outputs if the error is close to zero that means that from m feature vectors, which can be caluclated from the original, we can accurately determine the original n input vectors. As such instead of using n inputs we can use m since from m features we can get to n and back.

  It preserves the most important data that is needed for recunstruction(due to it yealding a lower error).

- Describe the role of the bottleneck in an autoencoder. Discuss how the size of the bottleneck can affect the model's reconstruction performance.

  The point of the bottle neck is to allow us to controll, the 'The dimensionality reduction' by allowing us to have a parameter for how many features we want in the latent space. If its size is too big, then the model will learn the identity function and will yeald no great use, if it is too small, it may be imposible for the model to accurately reconstruct data from the latent space as such we can't accurately sample from it.

- Explain what makes a variational autoencoder different from a standard autoencoder.

  Instead of learning how to calculate the features in the latent space, like a auto-encoder; a variational auto-encoder learns the distribution of each of the latent space features which allows us to sample from it.

- Name and briefly explain the purpose of the two parts of the loss term in the VAE loss function.

  The two parts of the loss are the recontruction term and the regularizer term. In a nutshell, we have two parts since our model has to learn two distinct 'things'. It needs to learn to reconstruct the input x' from the latent space (this is controlled by the reconstruction term), and it also needs to learn the distribution of each feature in the latent space, usually the normal distibution (this is controlled by the regularizer term).

- Discuss how a VAE can be used to generate new data points. What is the importance of the latent space distribution in this context?

  We can sample a random point from the normal distibution $N(0, 1)$ for each feature in the latent space. Our encoder learns the mean and standard distribution for each feature which we can use to 'reconstruct' it using the decoder. The importance of the distribution is that we can sample from it. If we didn't have a known distribution we would not be able to sample new features in the latent space and as such generate new data. It also makes sure that the regions are continuous/have a smooth transition as if there are holes, even if we pick a random point it is possible for the decoder to give useless results

3 Generative Adversarial Networks

- You settle on a GAN for your generative model. Describe the loss function for this model and how it is trained for the task of generating cats and dogs.

  The loss function is $-E_{x\ p_data} lnD(x) - E_{z\ p_z} ln(1 - D(G(z)))$. The loss function is simmilar to the cross-entropy and works in a simmilar way. To get A lower loss the model needs to correctly identify the real/training data as belonging in that group and simmilarly for fake/generated data.

  A GAN is composed of two networks. One network, called the discriminator; has the 'goal' of correclty identifying if the data is real or fake. Where as the generator has the goal of generating images, that the discriminator

would mark as real. For training first the discriminator is trained, then the generator (its error function is the opposite of the discriminator's). And we repeat until the equilibrium, where the discriminator can no longer determine if a generated sample is from the provided data set or not by using supervised learning.

- After training the network for some time, you observe that the model only generates images of huskies and a snowy background. What could have happened? Explain why is this an issue. Describe a strategy to mitigate this problem.

  The likely reason for this is that in the real data the only/majority of the data containing dogs contain huskeys. If it is the case, then the discriminator can easily distinguish which dog images are real and which are fake as such the error function of the generator that generates general dog images instead of just huskies is higher. One strategy to mitigate this is to simply increase the real data set to include a wider selection of dogs.

- After successful training, your model randomly provides you with either a cat image or a dog image; however, you are more likely to get a cat image. Explain why this could be the case.

  This could be the case, because of the imbalance of the training data. For instance the model might 'find it easier' to get cats marked as real than dogs due to the higher variety and as such will generate more cats(mode collapse) which could prevent it from generating a lot of dog pictures as well.

- For your intended use case, you want to be able to create a specific animal. Briefly discuss how you can adapt the approach to control which animal the model will generate.

  We can achieve this using Conditional GANs. Where the discriminator additionaly learns to detect if a certain atribute (for example is a cat) and then the generator learns to generate images with that attribute.

  (To expand the model to create more than two types of animals we can simply increase the real data set to include more animals).

4 Autoencoders

These are the graphs for changing the bottleneck for the AE.

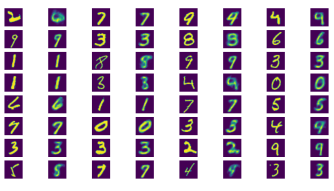

Figure 1: Training data and reconstruction having 10 as the bottleneck

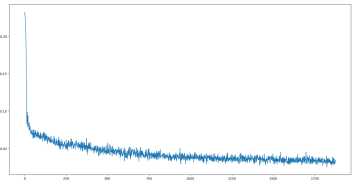

Figure 2: Training curve

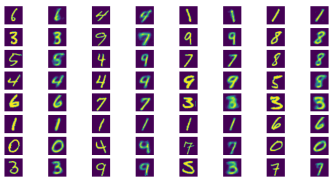

Figure 3: Training data and reconstruction having 20 as the bottleneck

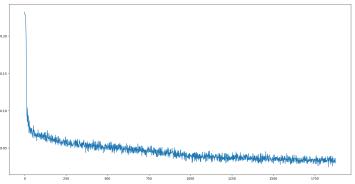

Figure 4: Training curve

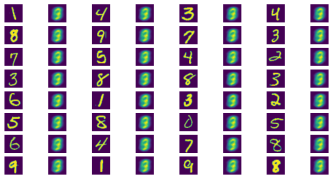

Figure 5: Training data and reconstruction having 2 as the bottleneck

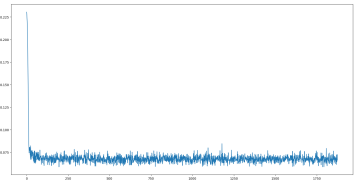

Figure 6: Training curve

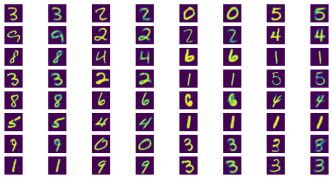

Figure 7: Training data and reconstr. with bottleneck of 10 and 20 epoches

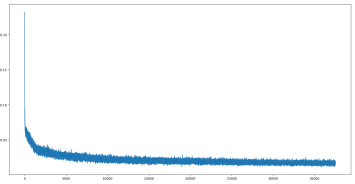

Figure 8: Training curve

Having 2 features in the latent space shows that it is insufficient to properly ‘describe‘ the images as all of the images have roughly the same blurred appearance

Having 10 features seems to preform quite nicely and, in some cases yealds a better looking number than the starting number (row 8 col 1), and in some cases yeald poor results (row 1 col 1). Increasing the epoch count increased the quality.

Having 20 features also yealds nice results, however the increase in accuracy seems almost negligable. All of these were trained on 1 epoch except for fig 4.

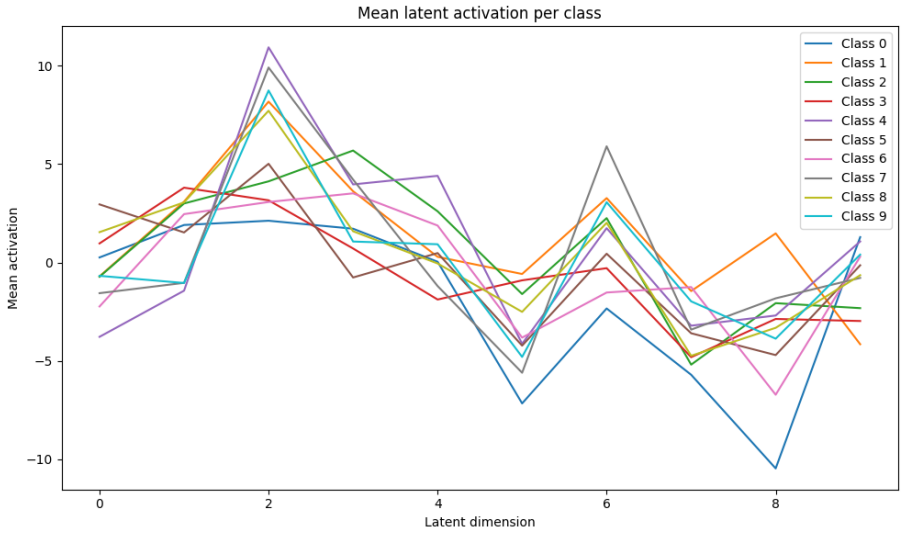

Figure 9: Mean activation graph

For low latent space sizes, the activations seem to be quite similar, where as with 8 activations it seems that all of the activations are quite unique.

5 Variational Autoencoders

The loss of the VAE consists of two parts, the reconstruction loss(which ensures that the reconstruction approaches the initial values) and the regularizer term(which ensures that the distribution modeled by the vae approaches a normal distribution). A reason as for why one loss may be weighed is because the reconstruction loss is averaged over all of the pixels, where ass the regularizer term is only for 10, as such the KL term may overpower the reconstruction loss.

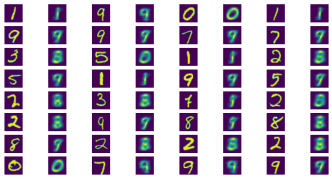

Figure 10: Training data and reconstr.

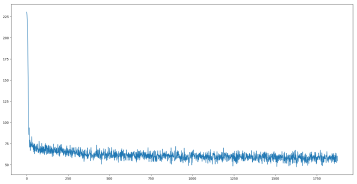

Figure 11: Total loss

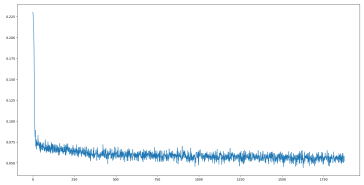

Figure 12: Reconstruction loss

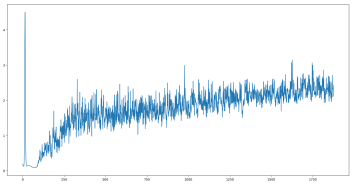

Figure 13: Regularization loss

The total loss and the reconstruction loss seems to increase as the training continues where as the regularitation loss seems to increase. This may be due to it having too strong of an impact in the beginning or due to the fact that to achieve better rezults, the distribution may need to sway a bit from a perfect normal distribution. The above model was trained on 8 latent space fields and 1 epoch below are trainings for 8 lat. sp. 10 ep., 4 lat. sp. 1 ep. and 4 lat. sp. 10 ep.

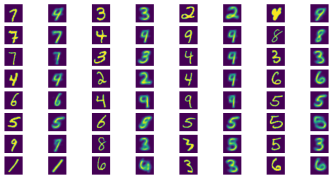

Figure 14: Training and reconstruction

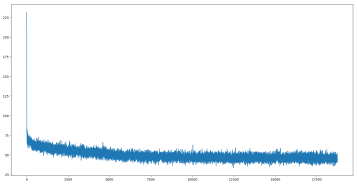

Figure 15: Total loss

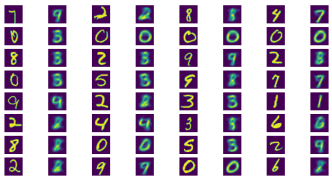

Figure 16: Training and reconstruction

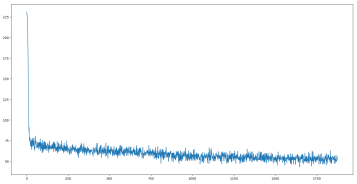

Figure 17: Total loss

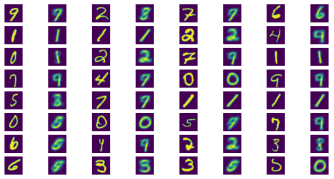

Figure 18: Training and reconstruction

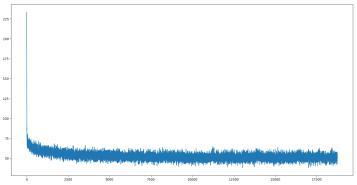

Figure 19: Total loss

For the model with 4 latent space elements, it seems as though, although additional epochs did help, some images are still more blurry/incorrect /4 lat. sp. 10 ep. row 1, col 1 looks more like a 7 than a 9)

Having 10 lat. sp. elements with one epoch still has ‘mistakes‘ (also row 1 col 1 looks like a 4) The one with 10 epochs looks better, although some images look better.

Below are the reencoded images via AE(20 lat. sp., 20 ep.)

Figure 20: Mean activation graph

Below are images generated by random sampling via the VAE(20 lat. sp., 10 ep.)

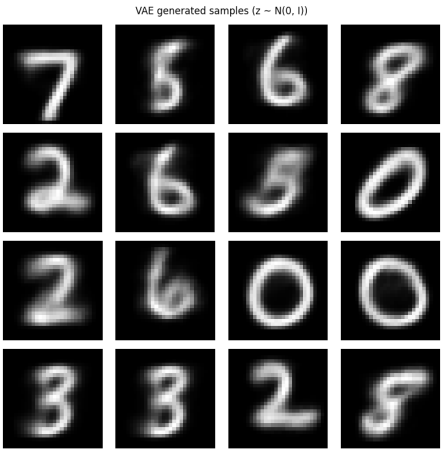

Figure 21: Mean activation graph

The reencoded images from the AE look more closely to the original training images, where as the VAE generated images, although some are a bit more blurry, do look more like standard images and do not look so strictly ‘handwritten‘ as in the first image and, in my opinion look pretty good :)