INT307 Multimedia Security System

Generative Learning

Shengchen Li

Xi'an Jiaotong-Liverpool University

07th Oct 2022

Aims

- Understand basic knowledge related to generative learning
- Know the basic procedure of media generation

Generative Model

- Describe how a dataset is generated with probabilistic model
- Usually the sampling process generates new media

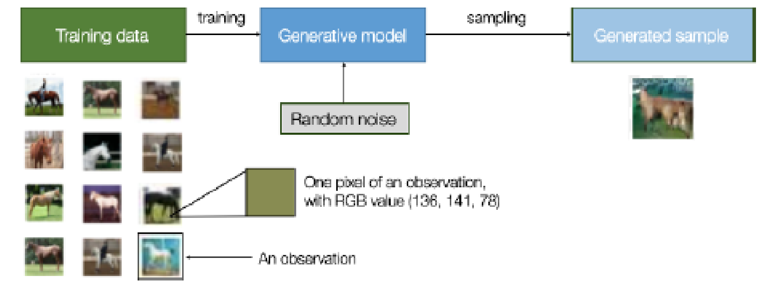


Figure 1-1. The generative modeling process

Feature Space and Likelihood

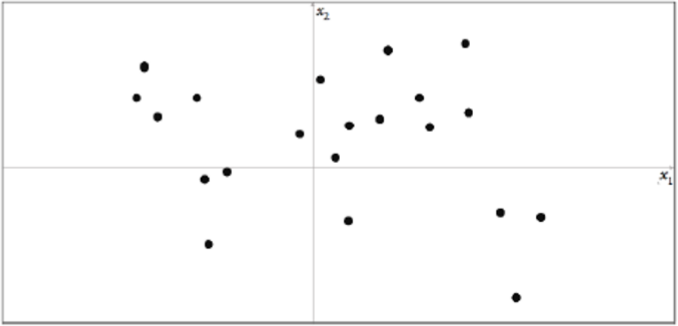


Figure 1-4. A set of points in two dimensions, generated by an unknown rule p_{data}

Autoencoder

- Input == Output
- Model structure could vary but be mirrored in most cases
- The middle layer (axis of mirroring) is usually considered as the resulted feature space
- If the dimension of feature space is fewer than original media, the autoencoder can be considered as a compressor

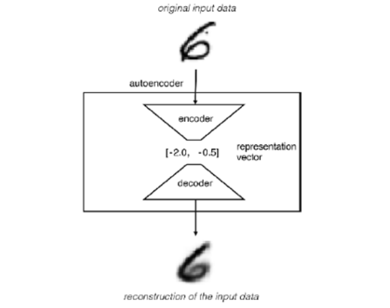


Figure 3-4. Diagram of an autoencoder

Sampling in Feature Space

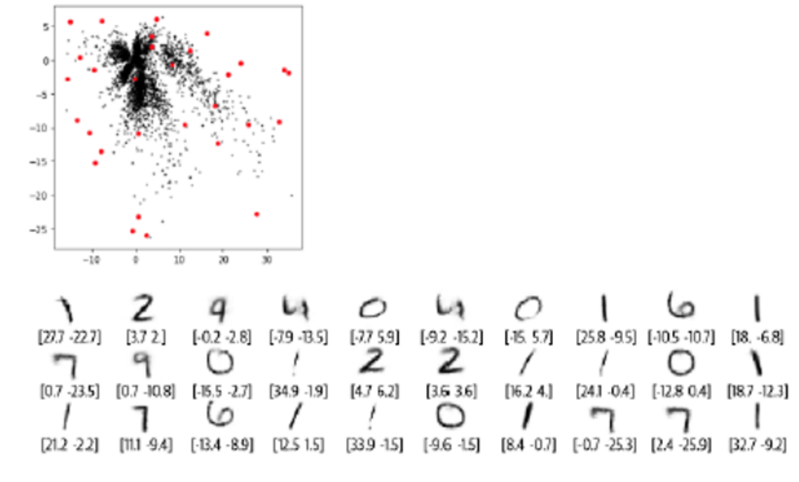


Figure 3-3. The new generative art exhibition

Problems with Autoencoder

- Feature space may not follow any particular probabilistic distribution hence it is impossible to do arithmetic operation in the feature space
- Similar features may not necessarily generate similar samples

VAE: Variational Autoencoder

- We force the feature space follows a Gaussian distribution / Gaussian Mixture Model
- A joint loss function is introduced, which is a sum of KL divergence between distributions and common reconstruction loss

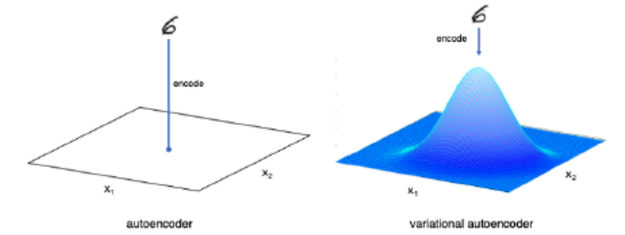


Figure 3-11. The difference between the encoder in an autoencoder and a variational autoencoder

VAE: Variational Autoencoder

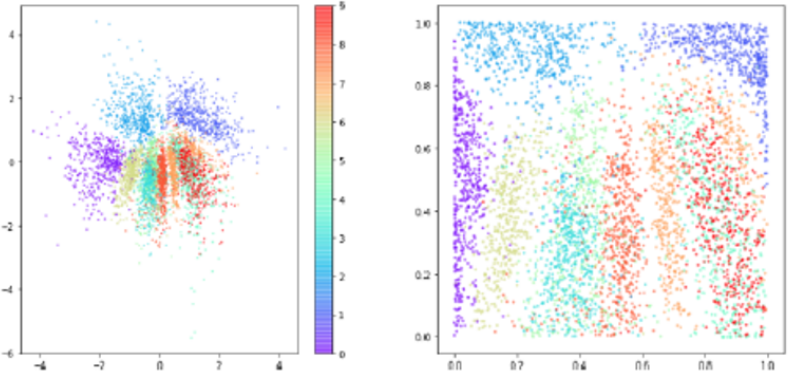
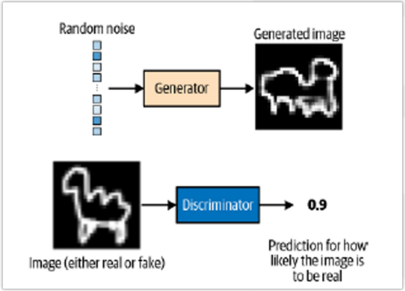


Figure 3-14. The latent space of the VAE colored by digit

Generative Adversarial Networks

- Generator: synthesise new samples
- Discriminator: judge whether the sample is synthesised



Training a GAN

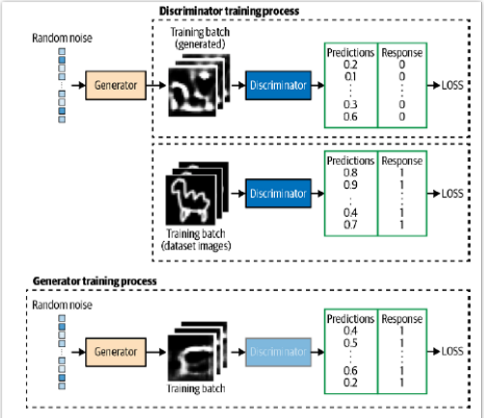


Figure 4-7. Training the GAN

Training a GAN

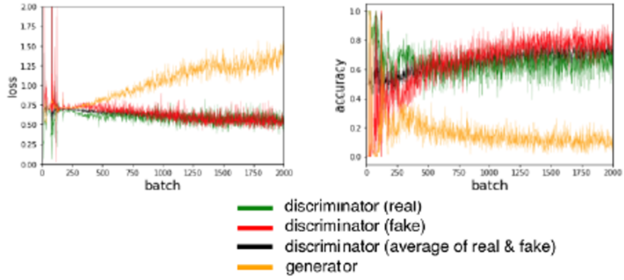


Figure 4-8. Loss and accuracy of the discriminator and generator during training

GAN Output

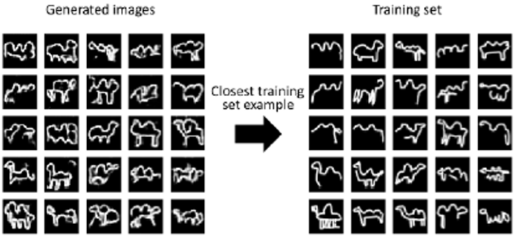


Figure 4-10. Closest matches of generated images from the training set

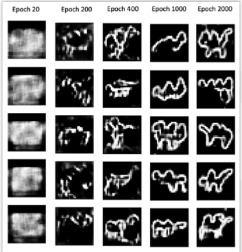


Figure 4-9. Output from the generator at specific epochs during training

GAN Challenges

Oscillating Loss

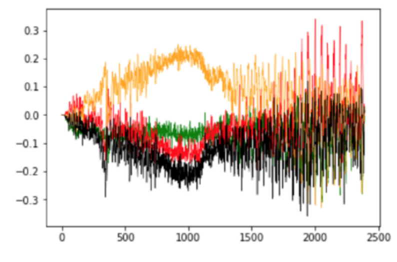


Figure 4-11. Oscillating loss in an unstable GAN

Model Collapse

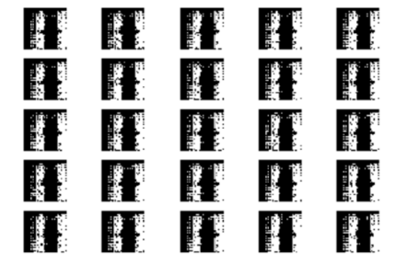


Figure 4-12. Mode collapse

Improvements on GAN

- WGAN (Wasserstein GAN)
  - Introduce Wasserstein loss
  - Lipschitz Constraint
  - Weight Clipping
- WGAN-GP: Apply gradient penalty loss

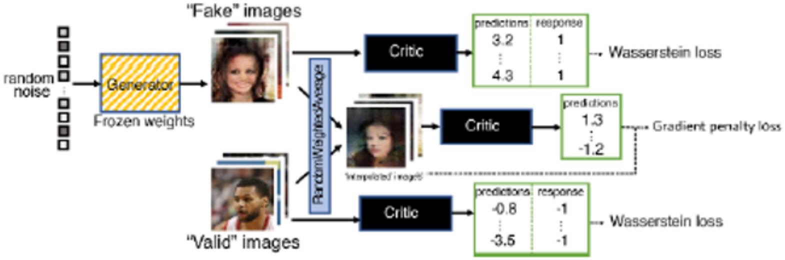


Figure 4-15. The WGAN-GP critic training process

WGAN analysis

Figure 4-14. Examples from the generator of a WGAN trained on images of horses

WGAN-GP Analysis

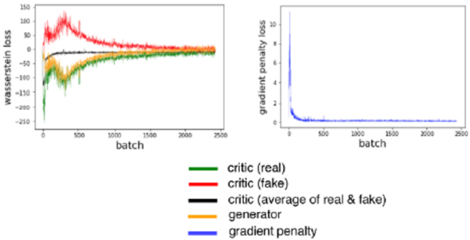


Figure 4-18. WGAN-GP loss

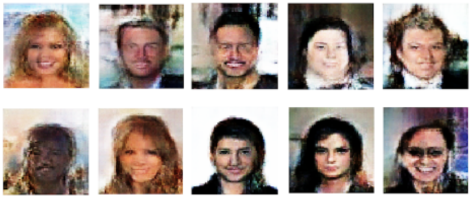


Figure 4-17. WGAN-GP CelebA examples

Face Generation (Deepfake)

Figure 1-3. Face generation using generative modeling has improved significantly in the last four years[4]

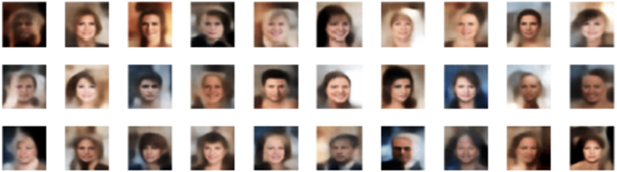


Figure 3-20. New generated faces

Feature Generation in Feature Space

Features can be generated by adding or subtracting feature vectors

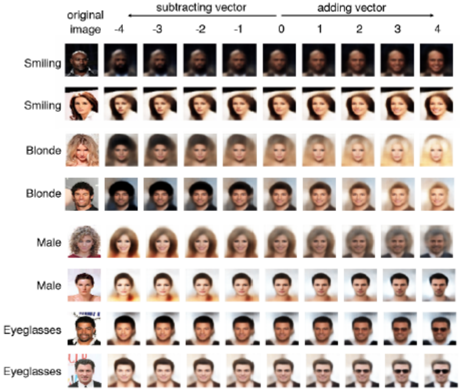


Figure 3-21. Adding and subtracting features to and from faces

Morphing between Faces

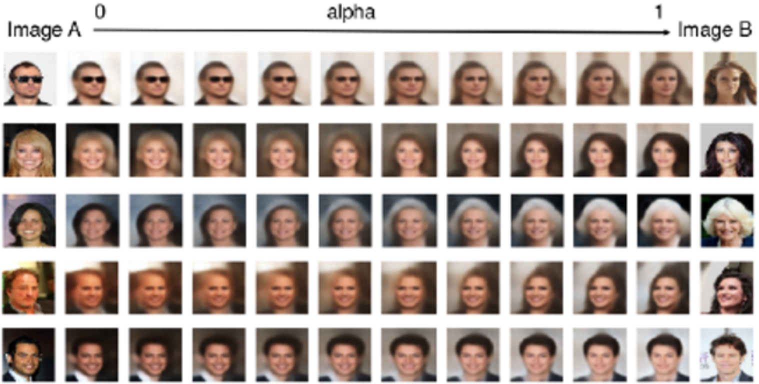
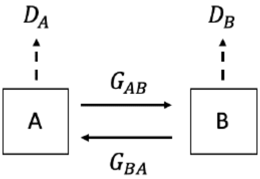


Figure 3-22. Morphing between two faces

Paints via Cycle GAN

- Two Generators: G_{AB} and G_{BA}
- Two Discriminators: D_A and D_B



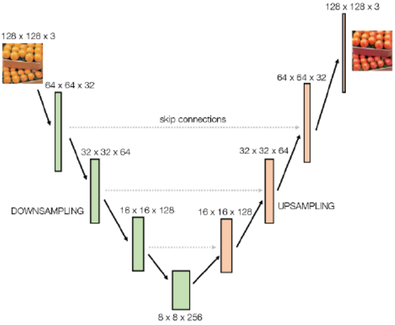


Figure 5-6. The U-Net architecture diagram

Cycle GAN

Figure 5-13. Output after 200 epochs of training[11]

Paints via Cycle GAN

- Content Loss
- Style Loss
- Total Variance Loss

Figure 5-14. An example of neural style transfer[13]

Figure 5-18. Output from the neural style transfer process at 1, 200, and 400 iterations

Write

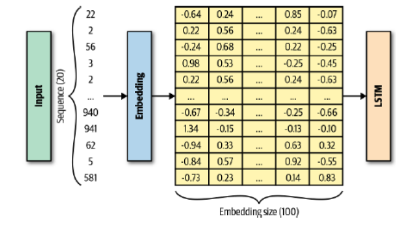


Figure 6-5. A single sequence as it flows through an embedding layer

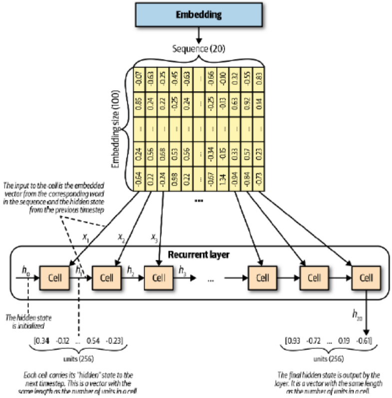


Figure 6-7. How a single sequence flows through a recurrent layer

Encoder – Decoder Structure

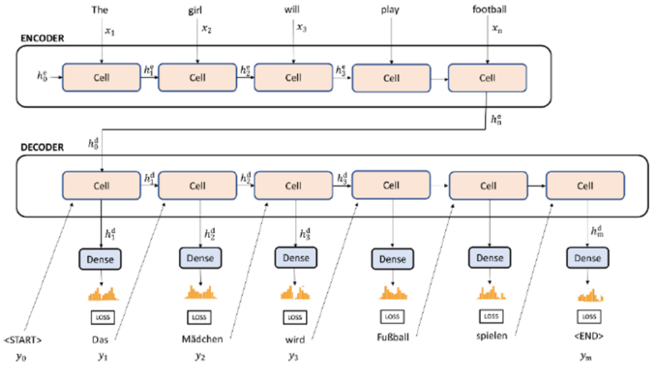
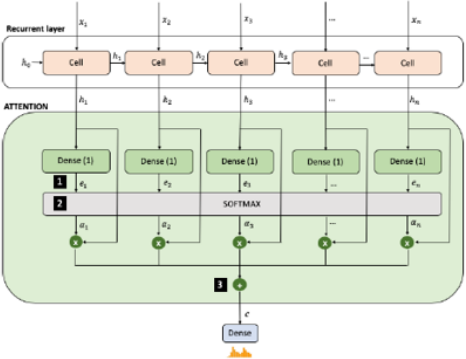


Figure 6-14. An encoder–decoder network

Attention Mechanism



1 $e_j = a(\boldsymbol{h}_j) = \tanh(W \cdot \boldsymbol{h}_j)$

2 $\alpha_j = [softmax(\boldsymbol{e})]_j = \dfrac{\exp(e_j)}{\sum_k \exp(e_k)}$

3 $c = \sum_j \alpha_j \boldsymbol{h}_j$

Figure 7-6. A recurrent layer for predicting the next note in a sequence, with attention

Composing with Attention

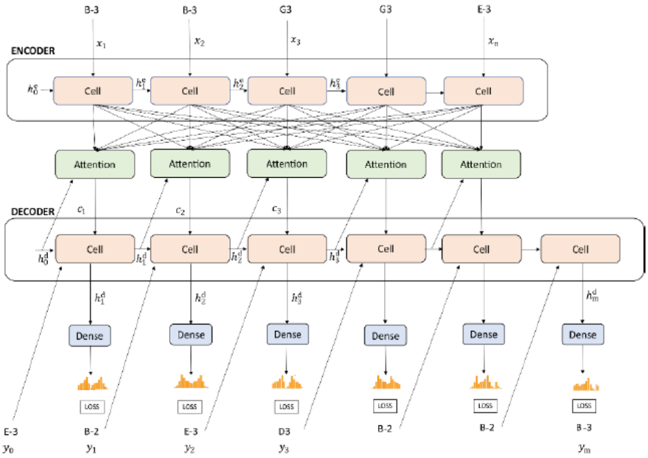


Figure 7-12. An encoder–decoder model with attention

Visualisation of Attention

Attention models the temporal relationship

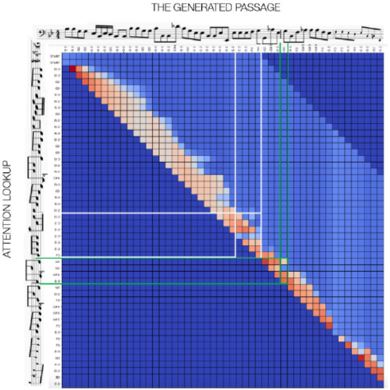


Figure 7-10. Each square in the matrix indicates the amount of attention given to the hidden state of the network corresponding to the note on the vertical axis, at the point of predicting the note on the horizontal axis; the more red the square, the more attention was given