ECE 8803-GGDL Homework 3

Arati Ganesh

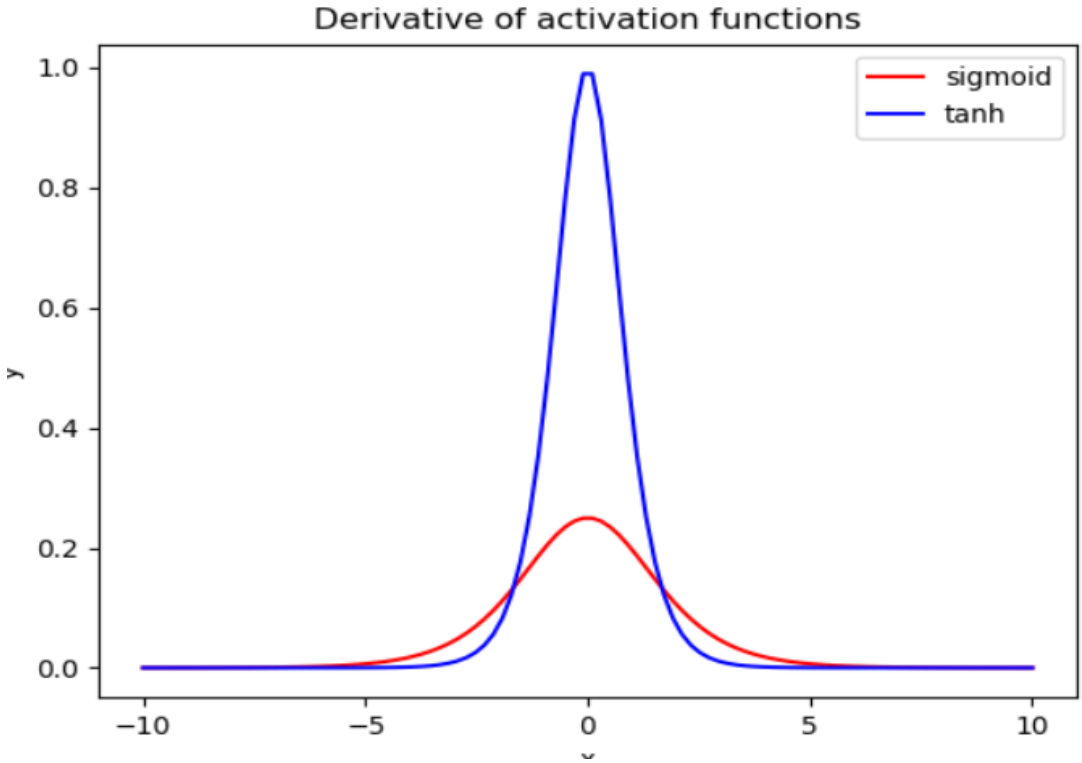
November 3, 2023

1. GANs on MNIST (50 points)

   You will implement a Generative Adversarial Network model and evaluate it under different conditions. You are provided with a notebook gan-mnist.ipynb, in which part of the necessary code has already been filled out for you. Pay attention to the cells marked `YOUR CODE HERE`: these are the only cells you need to edit. For those using the PACE clusters, a conda environment has been provided for you with the necessary packages for this notebook: hw3-p1.

   a. Why do you think it is convenient to normalize the images to the range $[-1, 1]$ instead of $[0, 1]$?

   Ans) Normalizing the data within the range [-1, 1] is done to achieve a zero-centered distribution. This alignment with a zero-centered distribution is crucial as it matches the characteristics of the tanh activation function employed in the generator's output layer. The preference for tanh over the sigmoid activation function is rooted in its slightly larger derivative, particularly in the proximity of zero. GANs are notorious for being challenging to train, often susceptible to issues like mode collapse or oscillations. The utilization of stronger gradients can act as a countermeasure to alleviate these problems by motivating the generator to explore a wider spectrum of potential samples.



---

1. Relation to Jensen-Shanon Divergence: How does the minimax game defined by the objective function

of a GAN relate to minimizing the Jensen-Shanon divergence between the training data distribution and the samples obtained from the generator? See https://arxiv.org/abs/1406.2661.

Ans) **Objective Function and Optimal Discriminator**

In the GAN framework, which consists of a generator (G) and a discriminator (D), the optimal discriminator $D^*G(x)$ for a given generator G is defined as:

$$D^*G(x) = \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_g(x)}$$

Objective Function for Discriminator

The discriminator's objective is to maximize the quantity $V(G, D)$ for any generator G, expressed as:

$$V(G, D) = \int p_{\text{data}}(x) \cdot \log(D(x))\, dx + \int p_g(x) \cdot \log(1 - D(x))\, dx$$

Reformulation and Training Objective

In GANs, the minimax game's objective is to find the generator G that minimizes the discriminator's ability to distinguish between real and generated data. This objective is expressed as $C(G) = \max_D V(G, D)$.

Connection to Jensen-Shannon Divergence

The global minimum of $C(G)$ is achieved when $p_g = p_{\text{data}}$. At this minimum, $C(G)$ reaches a value of $-\log(4)$. The objective function $C(G)$ can be expressed in terms of the Kullback-Leibler divergence (KL), resulting in:

$$C(G) = -\log(4) + 2 \cdot \text{JSD}(p_{\text{data}} \,||\, p_g)$$

This connection shows that the minimax game in GANs aims to train the generator to minimize the Jensen-Shannon divergence between the real data distribution (p_{data}) and the generated data distribution (p_g). The global minimum of $C(G)$ represents the scenario where the generator produces data that closely matches the data-generating process, indicating high-quality generated samples.

In conclusion, the GAN's minimax game and its associated objective function are fundamentally about minimizing the Jensen-Shannon divergence between the real data distribution and the distribution of generated samples.

---

2. Maximizing Incorrect Choices: Why is maximizing the probability of the discriminator making the INCORRECT choice the most practical way to update the generator? See the paper linked above.

Ans) The minimax equation in GANs, given by

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

is practical because it establishes a competitive framework between the generator (G) and discriminator (D).

The generator seeks to minimize its loss by producing data that is indistinguishable from real data ($\log(1 - D(G(z)))$), while the discriminator aims to maximize its accuracy in labeling real and generated data ($\log D(x)$). This competitive nature of the game provides clear and practical objectives for both

networks, resulting in stable and effective training. As a result, the generator progressively improves to generate high-quality and realistic data, making this approach a practical and powerful technique in GAN training.

---

d. Part 4: Training Function. Implement the optimizer construction, review and run the training loop cells. How do the generated samples look

Ans)

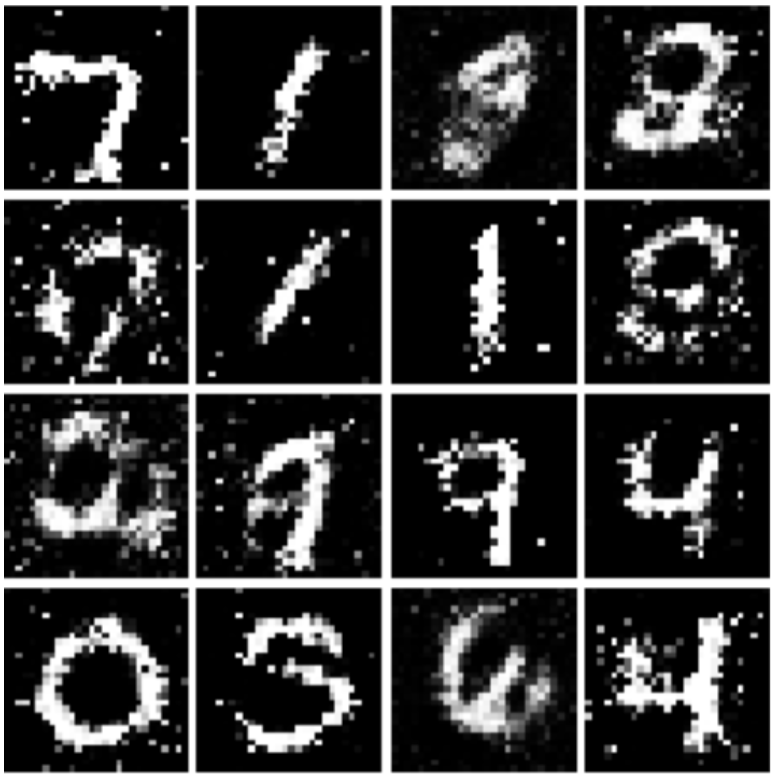


Figure 1: Vanilla GAN

Realism: The images produced by the Vanilla GAN often lack a high degree of realism. Some of the generated images fail to resemble recognizable digits and may exhibit distortions or artifacts.

Diversity: While the generated images encompass a range of modes from the dataset, there is a notable overemphasis on certain digits, particularly the digit ”1.” This overrepresentation of a single digit suggests a mode collapse, where the generator fixates on producing a limited set of samples and neglects the diversity in the data distribution.

Sharpness and Detail: The generated images generally lack the sharpness and fine detail characteristic of high-quality images. They may appear blurry or exhibit a lack of clarity, which impacts their overall visual quality.

---

e. Part 5: Least Squares GAN. Implement the Least-Squares loss function for the constructed networks and evaluate them. How does the resulting generated samples compare to those produced by the Vanilla GAN? Why do you think this is the case

Ans)

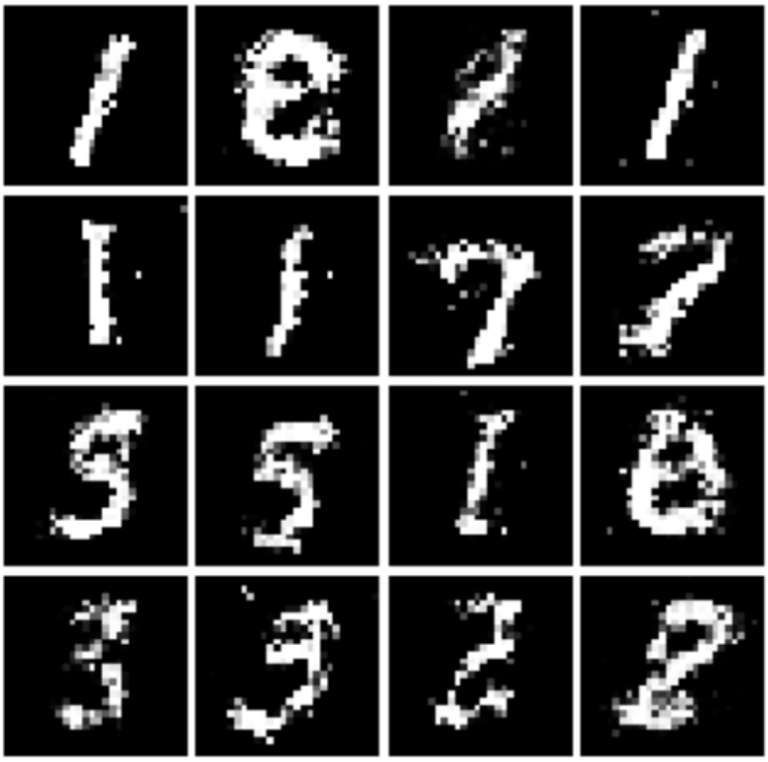


Figure 2: LS GAN

LSGAN surpasses Vanilla GAN in producing sharper and more discernible images due to its use of the L2 loss function. L2 loss ensures smoother gradients in the discriminator, which consider the distance to the decision boundary. This results in more informative gradients and better alignment with the data distribution. In contrast, Vanilla GAN relies on Binary Cross-Entropy loss, which saturates quickly and doesn't emphasize the distance from the boundary. Consequently, LSGAN's L2 loss leads to better image quality with sharper and more recognizable results.

---

f. Part 6: Deeply Convolutional GANs. Implement the generator and discriminator architec- tures including the convolutional layers specified in the notebook and evaluate the resulting model. How do the generated samples compare to the two previous configurations? Why does adding convolutional layers have this effect?

Ans)

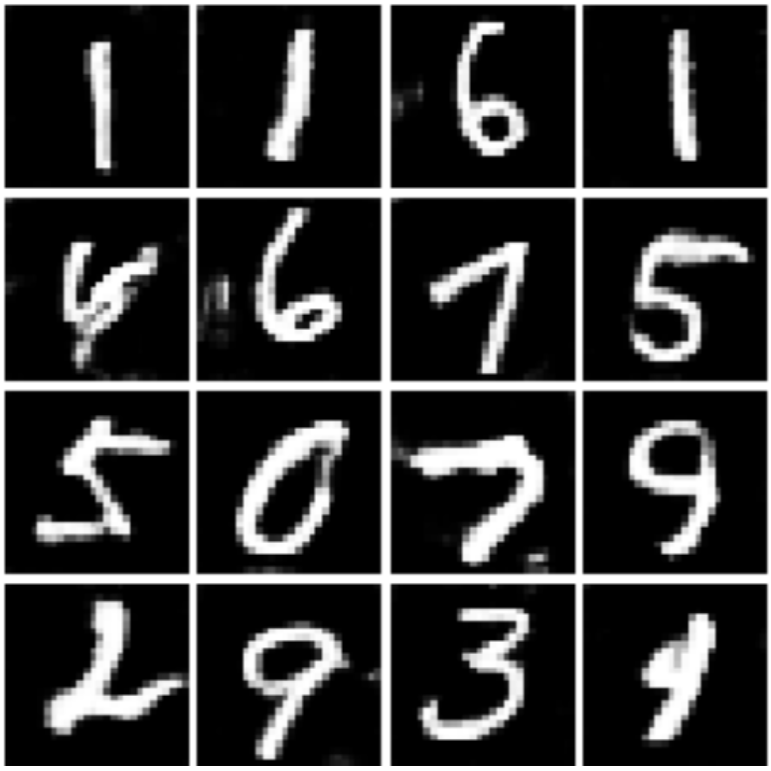


Figure 3: DC GAN

The generated image looks much more clean and natural compared to using Linear fully connected GANs. DCGAN typically outperforms Vanilla GAN and LS GAN due to its specialized architecture utilizing convolutional layers, batch normalization, and deep networks. This combination allows DCGAN to capture intricate spatial features and hierarchies, leading to the generation of sharper, more convincing, and higher-resolution images. Its stable training process and effective discriminator further contribute to its superior performance.

---

g. Part 7: Compare generated images. Evaluate the generated samples from the three previous models against the real MNIST images. Can you distinguish the original from the fake ones in all three cases? Do you think an equilibrium exists where the generator wins, i.e. the discriminator ends up unable to distinguish the two distributions on finite samples? Explain why you believe this would or would not be possible

- **Vanilla GAN:** Images generated by the Vanilla GAN may look somewhat realistic but can have distortions and blurriness. Human observers can often distinguish real from fake images with some accuracy.

- **LS GAN:** LS GAN produces sharper images compared to the Vanilla GAN but may still be distinguishable by keen observers.

- **DCGAN:** DCGAN is expected to produce high-quality images closely resembling real MNIST digits. Distinguishing real from fake images becomes increasingly challenging, and human observers may struggle to differentiate them.

Achieving a perfect equilibrium, where the discriminator cannot distinguish real from fake images under all circumstances, is challenging due to the complexity and variability of real data distributions.

An equilibrium where the generator consistently wins, rendering the discriminator unable to distinguish between the two distributions, is theoretically possible but often challenging to achieve on finite samples.

---

2. EBMs on MNIST (50 points). You will implement an Energy Based Model model and evaluate it for generative performance and anomaly detection capacity. You are provided with a notebook ebm-mnist.ipynb, in which part of the necessary code has already been filled out for you. Pay attention to the cells marked YOUR CODE HERE: these are the only cells you need to edit. For those using the PACE clusters, a conda environment has been provided for you with the necessary packages for this notebook: hw3-p2. a. Part 0/1: Libraries and utility functions. Read through the header sections of the notebook and run the corresponding cells to initialize the necessary libraries and functions (you do not need to edit any code here). How does the contrastive divergence objective change the energy of the correct/wrong samples? What do you think would need to happen for the push/pull forces on the diagram to balance out?

   Ans) The CD objective aims to lower the energy of correct samples. It encourages the model to assign lower energies to these samples, making them more likely states. For wrong samples, the CD objective seeks to raise their energy, making them less likely states. It adjusts the model's parameters to increase the energy of configurations that do not correspond to the training data.

---

b. Implement the model construction, MCMC sampling steps, and loss calculation in the corresponding cells and run the implemented model.

- **Swish vs. ReLU Activation**: Why do you think that the Swish activation function is more convenient than the standard ReLU for this model and task? Hint: Try to visualize the shape of both activation functions and think about how they would affect the (backward) propagation of the gradient through the network.
  Ans)
  Swish is used in energy-based models (EBMs) for two main reasons:

  1. Smoother Gradient: Swish provides a smooth and differentiable gradient, which ensures stable training and avoids the "dead ReLU" problem, benefiting the optimization process.

  2. Improved Training and Generalization: Swish has been shown to enhance training performance and generalization in EBMs, leading to better modeling of the underlying data distribution and more accurate energy landscapes.

- **Role of the Regularization Loss**: What is the role of the regularization loss? How do you think the resulting data would look if we were not using it?
  Ans) The role of the regularization loss in this context is to constrain the output values of the energy model (often represented as a neural network) to a reasonable range. Without the regularization loss, the output values could fluctuate over a very large range, which may lead to instability or difficulty in training.

  The regularization loss ensures that the values for the real data are around 0, and the fake data likely slightly lower. This constraint helps maintain stability in the training process and prevents the output values from going to extreme ranges.

If the regularization loss were not used, it is likely that the output values would become more erratic and less predictable. This could result in difficulties in convergence during training and may lead to the model generating less reliable or coherent data. The regularization loss helps maintain control over the output values, making the training process more stable and the generated data more consistent and meaningful.

---

c. Sample a few images using the MCMC method. How many MCMC steps are required to sample reasonable images from the trained model? Is this number different for different digits?

Ans) The number of MCMC steps required to sample reasonable images is 256. This is the same for different digits.

- **Random Noise Inputs**: For pure random noise inputs, are the scores in line with what would be expected for a good anomaly detection method? Why do you think the score for the 'true' images is so close to zero? Ans) Yes the the score considerably drops. Hence, the model can detect random Gaussian noise on the image. This is also to expect as initially, the "fake" samples are pure noise images.For true images the noise is zero and so the score is zero.
- **Image Transformations**: Do the scores of the transformed images differ significantly from the original ones? Do you think this model would make a good out-of-distribution detector? Provide a detailed explanation of why this would or would not be the case.

  The energy-based model effectively detects random noise, leading to a significant drop in scores for noisy images. However, for more complex transformations like flipping or resizing, the score differences are minimal, mainly due to the model's relatively small size. While it can be a good out-of-distribution detector for simple cases, its performance in detecting complex transformations depends on model depth and training duration. Deeper models can capture a wider range of variations and details, enhancing their out-of-distribution detection capabilities.
