Machine Learning & Deep Learning

Tutorial

MSCV/ESIREM

Antoine Lavault
antoine.lavault@u-bourgogne.fr

VAE, GANs and Similarity learning

Any question or exercise marked with a "*" is typically more technical or goes further into developing the tools and notions seen during class.

Problem 1

Basic concepts.

1. Describe quickly how a (deterministic) auto-encoder works. Same question with a variational auto-encoder. A deterministic autoencoder and a variational autoencoder (VAE) are both neural network architectures for slightly different use cases. Both work with a notion of encode-decode into a bottleneck layer.

   A deterministic autoencoder learns a fixed, deterministic mapping from input to latent representation. In contrast, a variational autoencoder introduces probabilistic elements, making it suitable for generating new data samples from the learned distribution in the latent space.

   VAEs are often used in generative tasks like image generation and data synthesis, whereas deterministic autoencoders are primarily used for dimensionality reduction and denoising.

2. Describe briefly how Generative Adversarial Networks (GAN) work.

   Generative Adversarial Networks (GAN) consist of two neural networks: a generator and a discriminator. The generator generates data from random noise (the latent), while the discriminator attempts to distinguish real data from fake data.

   Through adversarial training, the generator aims to produce data that is indistinguishable from real data, leading to the generation of realistic and high-quality data samples. In adversarial training, both networks are set in opposition to one another, each having opposing objectives, yet they must still learn from and adapt to the other's feedback. However, stability is not guaranteed.

   GANs are widely used in tasks like image generation and style transfer.

3. Describe briefly what similarity learning is.

   Similarity learning is a machine learning approach focusing on training models to understand and quantify the similarity or dissimilarity between data points. It aims to represent data where similar items are closer in the feature space, making comparing and classifying them easier. This technique is commonly used in recommendation systems, image retrieval, and face recognition to identify and group similar items or objects based on their characteristics or features.

Problem 2

VAE https://www.overleaf.com/project/6515187d975add6955ddcec2
Check the last tutorial for an exercise on this.

Problem 3

Condition of optimality for a GAN We will prove here some results found in [Goodfellow et al., 2014].

1. Recall the GAN loss function. See below.

2. The generator loss can "saturate" during the early phases of the training process. What could be the reason, and what improvement can be made to the optimization process to avoid this problem? In the context of a Generative Adversarial Network (GAN), "saturation" typically refers to a problem where the discriminator's output saturates or becomes stuck at extreme values (e.g., 0 or 1), which can stall the training of the GAN.

   Saturation especially occurs when the discriminator loss is close to 0 or 1 for a substantial portion of the training data. When the discriminator confidently assigns a probability close to 0 or 1, it provides very little gradient information for the generator to learn from. This effectively causes the training of the generator to stop.

   Generally, a saturation of 0 of the generated samples means the discriminator wins. The other way around means the discriminator is too weak.

Global Optimality of $p_g = p_{data}$. The generator G implicitly defines a probability distribution p_g as the distribution of the samples $G(z)$ is obtained when $z \sim p_z$. Therefore, we would like the GAN training algorithm to converge to a good estimator of p_{data} if given enough capacity and training time.

Recall that the GAN training criterion is:

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

1. Show that the function $y \mapsto a \log(y) + b \log(1 - y)$ achieves its maximum in $[0, 1]$ at $\frac{a}{a+b}$. Not too hard. Right?

2. We first consider the optimal discriminator D for any given generator G. Write the training criterion V as a set of integrals. The expectation is integral.

   $$\int_x p_{data}(x) \log(D(x))dx + \int_z p_z(z) \log(1 - D(G(z)))dz$$

   But p_z and p_g are somewhat close:

   $$\int_x p_{data}(x) \log(D(x))dx + \int_x p_g(x) \log(1 - D(x))dz$$

3. Deduce an expression for the optimal discriminator D^*_G given an arbitrary generator G.

   For a given G, we optimize D with $C(G) = \max_D V(G, D)$.

   We use the previous questions to end up with $\frac{p_{data}}{p_{data}+p_g}$

4. Given the integral equation obtained above, rewrite the training criterion for the discriminator $C(G) = \max_D V(G, D)$ function of p_{data} and p_g

   Note that the training objective for D can be interpreted as maximizing the log-likelihood for estimating the conditional probability $P(Y = y|x)$, where Y indicates whether x comes from p_{data} (with $y = 1$) or from p_g (with $y = 0$).

   $$E_{x\sim p_{data}}[\log D^*(G(x))] + E_{x\sim p_g}[log(1 - D^*(G(x))0)]$$

   And we know the expression of D^*

5. We will show that the global minimum of the virtual training criterion $C(G)$ is achieved if and only if $p_g = p_{data}$. $C(G)$ achieves the value $-\log 4$ at that point.

   (a) Show that when $p_g = p_{data}$, we have $D^*_G(x) = \frac{1}{2}$. Then, deduce that we have $C(G) = -\log 4$ Using the answer above, we have $2\log 1/2 = -\log 4$

   (b) Recall the expression of the Kullback-Leibler divergence. Calculate its value for p_g and p_{data}. It looks a lot like the expression of $C(G)$ in a way.
   And the other way round $E_{p_{data}}[\log p_{data}/p_g]$

   (c) Show that $C(G) = -\log(4) + 2JSD(p_{data} \parallel p_g)$.

   (d) How to interpret this result? When everything is alright, a GAN optimizes the Jensen-Shannon divergence between the real and fake data distribution, matching the distributions.

Problem 4

Evolution of GANs (*)

LSGAN Least Square GAN [Mao et al., 2017] (LSGAN) optimizes the following :

$$\min_D V_{LSGAN}(D) = \frac{1}{2}E_{x\sim p_{data}}(x)[(D(x) - b)^2] + \frac{1}{2}E_{z\sim p_z}(z)[(D(G(z)) - a)^2]$$

$$\min_G V_{LSGAN}(G) = \frac{1}{2}E_{z\sim p_z}(z)[(D(G(z)) - c)^2]$$

We are assumed to use the $a - b$ coding scheme for the discriminator, where a and b are the fake and real data labels, respectively. c denotes the value G wants D to believe for the fake data.

1. Show that for a fixed G, the optimal discriminator D is

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

   Use the same trick as question 2 in "Global optimality of GANs."

   Then, minimize $1/2(p_{data}(x)(D(x) - b)^2 + p_g(x)(D(x) - a)^2)$. with respect to $D(x)$.

2. Start by remarking that adding a term $E_{x\sim p_{data}}(x)[(D(x)-c)^2]$ to $V_{LSGAN}(G)$ causes no change in the optimal values. Show that:

$$2C(G) = \int_{\mathcal{X}} \frac{((b-c)(p_{data}(x)+p_g(x)) - (b-a)p_g(x))^2}{p_{data}(x)+p_g(x)} dx.$$

First, write the equation with the hint. And then find it looks a lot like an expectancy, so integrals.

3. Show that optimizing LSGANs yields minimizing the Pearson χ^2 divergence between $p_{data}+p_g$ and $2p_g$, if a, b, and c satisfy the conditions of $b-c=1$ and $b-a=2$. We define the Pearson χ^2 divergence as:

$$\chi^2_{Pearson}(P \parallel Q) = \int \frac{(p(x)-q(x))^2}{q(x)} dx$$

Write the Pearson divergence and then the rest.

The Kullback-Leibler divergence is widely used in variational inference due to the convenient evidence lower bound (ELBO, see VAE), and the Jensen-Shannon divergence has very similar properties. However, optimizing $KL(p_g \parallel p_d)$ has the problem of mode-seeking behavior or under-dispersed approximations. This problem also appears in GANs learning, known as the mode collapse problem. In other words, the GAN learns to reproduce only one or a few modes of the training data.

The advantage of using the Pearson χ^2 divergence instead of the Kullback-Leibler/Jensen-Shannon divergence is that it makes LSGANs less mode-seeking and alleviates the mode collapse problem.

Reading exercise - Wasserstein GAN This exercise is a *reading* exercise. The questions are only here to guide you somewhat to extract the information and make your conclusions.

Wasserstein GAN uses the Wasserstein distance (a distance between probability distributions) instead of the KL or Pearson divergence.

1. Find the definition of the Wasserstein distance and the Kantorovich-Rubenstein duality theorem. Note the special hypothesis for the optimization in the duality theorem.

2. Find the training algorithm for the Wasserstein GAN From the original article. What is striking with the new loss compared to the original GAN or LSGAN?

3. The original version uses weight clipping to enforce a special hypothesis. What is this special hypothesis, and is clipping a good thing?

4. An improvement was found in [Gulrajani et al., 2017] but there is a problem. Which one?

5. Propose an improvement and compare it to what was introduced in [Adler and Lunz, 2018].

Optimization over 1-Lipschitz functions. Weight clipping is a dirty way of doing it. WGAN-GP is too strong (forces 1-lipschitz functions only), and finally, WGAN-LP works as intended.

Problem 5

Reading exercise - GAN in audio: DrumGAN vs. StyleWaveGAN This exercise is a *reading* exercise. The questions are only here to guide you somewhat to extract the information and make your conclusions. Also, these articles are not about vision or robotics to push you outside of your comfort zone. This might also imply you will need to read some of the bibliography.

1. Prerequisite: Read [Karras et al., 2018] and [Karras et al., 2020]
2. Make a table identifying the common points and the differences between DrumGAN [Nistal et al., 2020] and StyleWaveGAN [Lavault et al., 2022].
3. One of the common points is the use of perceptual controls in the synthesis. Describe in more detail what are the similarities and differences between the approaches.
4. A staple in model evaluations, especially in audio, is subjective evaluation. Only one article explores the just-noticeable difference in perceptual controls with GANs. Find it.

PGAN and StyleGAN are two big steps in GAN development. For the rest, do the work. It's only reading.

And I'm slightly biased.

References

[Adler and Lunz, 2018] Adler, J. and Lunz, S. (2018). Banach Wasserstein GaN. In *Advances in Neural Information Processing Systems*, volume 2018-Decem, pages 6754–6763.

[Goodfellow et al., 2014] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., and Weinberger, K., editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.

[Gulrajani et al., 2017] Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. (2017). Improved training of wasserstein GANs. In *Advances in Neural Information Processing Systems*, volume 2017-Decem, pages 5768–5778.

[Karras et al., 2018] Karras, T., Aila, T., Laine, S., and Lehtinen, J. (2018). Progressive growing of GANs for improved quality, stability, and variation. In *International Conference on Learning Representations*.

[Karras et al., 2020] Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., and Aila, T. (2020). Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8110–8119.

[Lavault et al., 2022] Lavault, A., Roebel, A., and Voiry, M. (2022). StyleWaveGAN: Style-based Synthesis of Drum Sounds with Extensive Controls using Generative Adversarial Networks. In *19th Sound and Music Computing Conference (SMC 2022)*, Saint-Etienne, France.

[Mao et al., 2017] Mao, X., Li, Q., Xie, H., Lau, R. Y., Wang, Z., and Paul Smolley, S. (2017). Least squares generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2794–2802.

[Nistal et al., 2020] Nistal, J., Lattner, S., and Richard, G. (2020). Drumgan: Synthesis of drum sounds with timbral feature conditioning using generative adversarial networks. In *ISMIR*.