Generative Adversarial Networks for MNIST

Thomas Boudras
Mehdi Inane
Augustin Gervreau

LAB2 Gan's

PSL - IASD

November 17, 2023

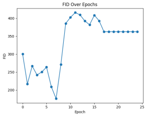
Vanilla GAN

- Training the generator every n epochs
- Discriminator with dropouts
- Batch normalization

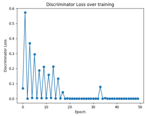
- Training the generator less often than the discriminator leads to a more stable training
- Lower dropout improved the overall performance of the model
- Batch normalization didn't have much effect on the training

Figure 1: Every iteration training

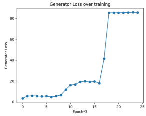
Figure 2: training the generator every 3 iterations



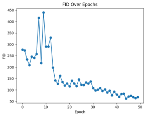
(a) Vanilla n = 1, FID



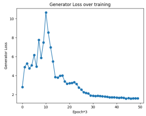
(b) Vanilla n = 1, D loss



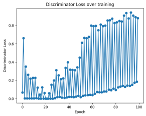
(c) Vanilla n = 1, G loss



(a) Vanilla n = 2, FID



(b) Vanilla n = 2, D-loss



(c) Vanilla n = 2, G-loss

WGAN

- In vanilla GAN, we want to optimize :

$$min_G max_D \mathbf{E}_{x \sim P_r}[log(D(x))] + \mathbf{E}_{\widetilde{x} \sim P_g}[1 - log(D(\widetilde{x})] \qquad (1)$$

  where P_r and P_g denote respectively the real / generated distributions and the output of D is a probability.

- On the other hand, the Wasserstein-1 GAN objective function becomes [3]:

$$min_G max_{D \in \hat{D}} \mathbf{E}_{x \sim P_r}[D(x)] + \mathbf{E}_{\hat{x} \sim P_g}[1 - log(D(\hat{x})] \qquad (2)$$

- The difference here is in the output of D, that becomes a score, and in the fact that $\hat{x} = \epsilon x + (1 - \epsilon)\widetilde{x}$. Note that we would like to restrain D to $\hat{D}$, which is the set of 1-Lipschitz discriminators

WGAN

- Neural networks can approximate many functions so can find appropriate 1-Lipschitz functions. (Universal approximation theorem).
- In practice we do not have a huge model. And constraining the model to choose such a function is the problem here.
- The initial papers came with the idea of weight clipping: all parameters are constrained to $[-c, c]$ [2]
- Problem is the loss of expressivity of D which is too constrained (learns very simple functions). Training the Generator on this does not work very well, especially for slightly wrong c.
- In theory this way often leads to vanishing or exploding gradients [3].

Improved WGAN

- A new way to enforce the constrained was proposed: Gradient Penalty[3]. Stop clipping and regularize D loss with $\mathbf{E}_{\hat{x}\sim P_{\hat{x}}}[(||\nabla_{\hat{x}} D(\hat{x})||_2 - 1)^2]$ (where $P_{\hat{x}}$ the uniform distribution on the straight line between samples.
- Problem observed in [5] is that this is unable to bound the gradient almost everywhere. Doesn't explore whole support entirely. New proposition: CP (perturb real data point: penalty on gradient around manyfold $x \sim \mathbb{P}_r$)
- Finally we found several other flavours all attempting to improve on this problem, such as [1] and [4]. We plan on exploring some of those to compare the result and see what is best.

Some results on WGANs

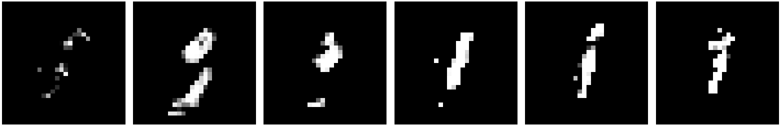


Figure 5: Output of GP-WGAN on one fixed vector in the latent space throughout epochs.

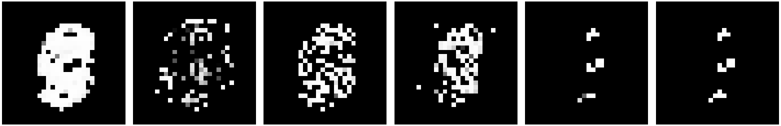
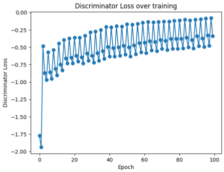


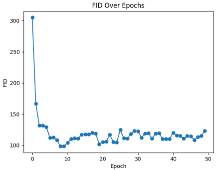
Figure 6: Output of WGAN with weight clipping on one fixed vector in the latent space throughout epochs.

Some results on WGANs

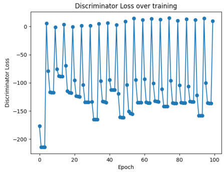


(a) Discriminator Loss (GP)

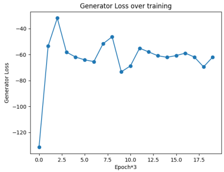
(b) Generator Loss (GP)



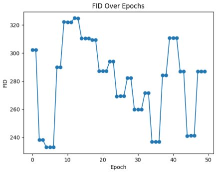
(c) FID Plot (GP)



(a) Discriminator Loss (clipping)



(b) Generator Loss (clipping)



(c) FID Plot (clipping)

Varying the regularization parameter on the gradient

(a) Discriminator Loss (b) Generator Loss (c) FID Plot

Figure 9: Training the WGAN with smaller regulizer term with $\lambda = 3$

Discriminator Driven Latent Sampling (DDLS)

- Improve on the generator of pretrained GAN
- Define boltzman distribution $p_d^* = e^{log(p_g(x)+d(x))}/Z_0$
- Define the energie $E(z) = -log(p_0(z) - d(G(z))$ and its Boltzman distribution $p_t(z) = e^{-E(z)}/Z$
- Then we can prove that : when D is the optimal discriminator, $p_d^* = p_d$ and if $z \sim p_t$, then $G(z) = x \sim p_d^*$
- We sample from this Boltzmann distribution p_t with an MCMC sampler : $z_{i+1} = z_i - \frac{\epsilon}{2}\nabla_x E(x) + \sqrt{\epsilon}n$, with $n \sim \mathcal{N}(0, I)$
- Possible merger of WGAN and DDLS methods

Bibliography

[1] Jonas Adler and Sebastian Lunz. "Banach Wasserstein GAN". In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio et al. Vol. 31. Curran Associates, Inc., 2018.

[2] Martin Arjovsky, Soumith Chintala, and Léon Bottou. *Wasserstein GAN*. 2017. arXiv: 1701.07875 [stat.ML].

[3] Ishaan Gulrajani et al. "Improved Training of Wasserstein GANs". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017.

[4] Huidong Liu, Xianfeng Gu, and Dimitris Samaras. "Wasserstein GAN With Quadratic Transport Cost". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Oct. 2019.

[5] Xiang Wei et al. *Improving the Improved Training of Wasserstein GANs: A Consistency Term and Its Dual Effect*. 2018. arXiv: 1803.01541 [cs.CV].