# Assignment3 Report: Generative Adversarial Networks

CS5242 Neural Network and Deep Learning


Meng-Jiun Chiou

mengjiun.chiou@u.nus.edu


November 21, 2017

## 1 Question 1

Given the loss function $C(w, b, \theta)$, the gradients of $w$ can be calculated as follows:

$$\frac{\partial C(w,b,\theta)}{\partial w} = \frac{-x\exp(-wx-b)}{(1+exp(-wx-b))^2} + \frac{\theta z\exp(-w\theta z-b)}{(1+\exp(-w\theta z-b))^2}. \tag{1}$$

With simple analysis, we can find that the cost $C(w, b, \theta)$ increases as the value of $|\theta z - x|$ increase when $\theta z - x > 0$ and the cost decreases when $\theta z - x < 0$. As $w \in [-1, 1]$,

$$w = \begin{cases} -1 & \text{, if } \theta z > x \\ 1 & \text{, if } \theta z < x \end{cases}. \tag{2}$$

The gradient of $b$ w.r.t. $C(w, b, \theta)$ can be calculated as follows:

$$\frac{\partial C(w,b,\theta)}{\partial b} = \frac{-\exp(-wx-b)}{(1+exp(-wx-b))^2} + \frac{\exp(-w\theta z-b)}{(1+\exp(-w\theta z-b))^2}. \tag{3}$$

To calculate the value of $b$ that results to global minimum, let the gradient $\frac{\partial C(w,b,\theta)}{\partial w} = 0$ and we can get the value of $b$:

$$b = \frac{-w(\theta z + x)}{2}. \tag{4}$$

The gradient of $C(w, b, \theta)$ with respect to $\theta$ is:

$$\frac{\partial C(w,b,\theta)}{\partial \theta} = \frac{-wz\exp(-wx-b)}{(1+exp(-wx-b))^2} \tag{5}$$

When $w \geq 0$, to make loss lower, the value of $\theta$ tends to be higher, it becomes $\frac{\beta}{z}$ (upper bound) in this example. Otherwise when $w < 0$, $\theta$ becomes $\frac{\alpha}{z}$ (lower bound).

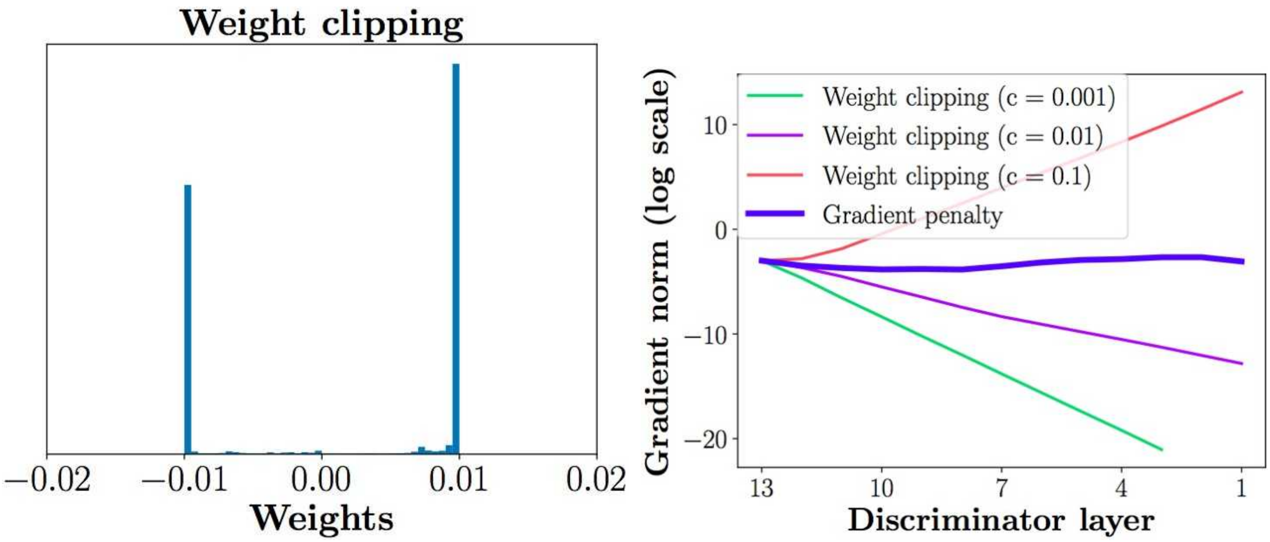
$$\theta = \begin{cases} \frac{\beta}{z} & \text{, if } w \geq 0 \\ \frac{\alpha}{z} & \text{, if } w < 0 \end{cases}. \tag{6}$$

For $\theta z < x$, the $(w, \theta)$ becomes $(1, \frac{\beta}{z})$ after the first iteration. Since the change of $\theta$, it results in $\theta z > x$ in the second iteration. $(w, \theta)$ thus becomes $(-1, \frac{\alpha}{z})$. As one might imagine, $(w, \theta)$ will then bounce between $(1, \frac{\beta}{z})$ and $(-1, \frac{\alpha}{z})$. This is why GAN does not converge.

## 2 Question 2

The problem of Wasserstein GAN may comes from the weight clipping part. The weight clipping is at the final step in each iteration.

$$\leftarrow clip(w, -c, c) \tag{7}$$




(a) Centralized weights at maxima and minima.

(b) Gradient exploding/vanishing as the $c$ is too big/small

Figure 1: The bad effects caused by weight clipping

The equation 7 clips the weight within $[-c, c]$ ($c = 0.01$ for example). However, it may causes the weights to concentrate to either point ($-0.01$ or $0.01$), as show in figure 1a [1].

On the other hand, the constant $c$ in weight clipping may causes gradient exploding if the constant is too big, or gradient vanishing if the constant is too small, as show in figure 1b [1].

Gradient penalty [2] has been proposed to deal with this problem. To improve, they remove the weight clipping part but add another loss to limit the gradient of discriminative model. The final loss is as follows: As

$$L(D) = -\mathbb{E}_{x\sim P_r}[D(x)] + \mathbb{E}_{x\sim P_g}[D(x)] + \lambda\mathbb{E}_{x\sim \mathcal{P}_{\hat{x}}}[||\nabla_x D(x)||_p - 1]^2$$

showed in figure 2, the revised version of WGAN can improve slow convergence problem in original WGAN.

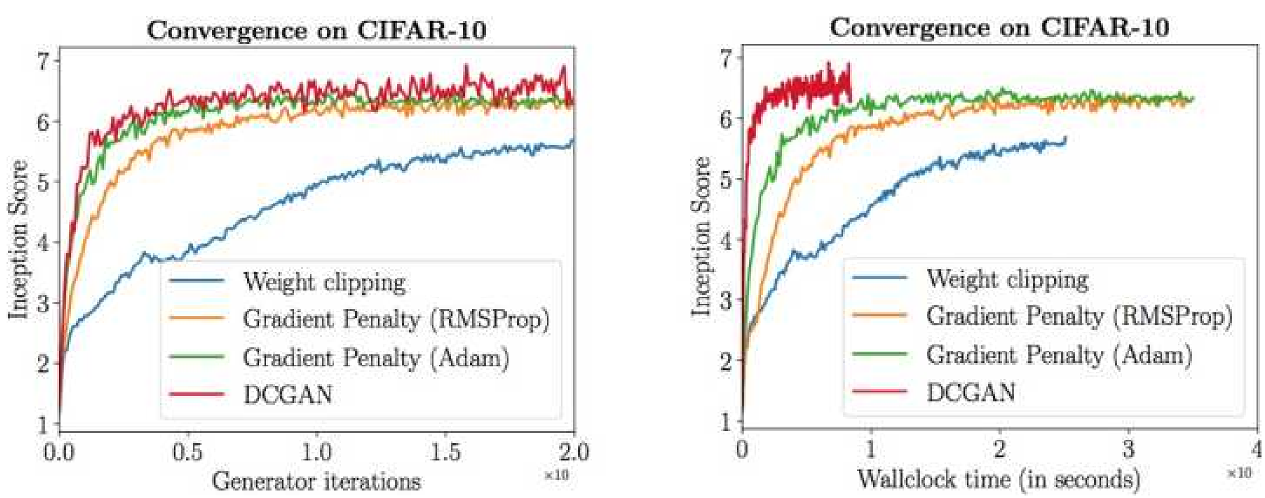



Figure 2: The convergence comparison of improved WGAN to WGAN and DCGAN.

# References


[1] "Improved wgan explanation." `https://www.zhihu.com/question/52602529`. Accessed: 2017-11-21.

[2] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, "Improved training of wasserstein gans," *arXiv preprint arXiv:1704.00028*, 2017.