# Paper

- **Title:** On the Effects of Batch and Weight Normalization in Generative Adversarial Networks
- **Authors:** Sitao Xiang, Hao Li
- **arXiv link:** https://arxiv.org/abs/1704.03971

# TL;DR

The paper presents a weight normalization technique for training GANs that improve the stability of the training and also a squared Euclidean reconstruction error on a test set to objectively assess performance of a GAN.

# Weight Normalization

Salimans and Kingma (2016) suggested the following weight normalization:
For a linear layer, $y = W^T x + b$, where $x \in \mathbb{R}^n$, $y \in \mathbb{R}^m$, $W \in \mathbb{R}^{n \times m}$ and $b \in \mathbb{R}^m$, reparameterize $W$ as with $V \in \mathbb{R}^{n \times m}$ and $g \in \mathbb{R}^m$ as:

$$w_i = \frac{g_i}{||v_i||_2} v_i$$

where $w_i$ and $v_i$ are the $\mathrm{i}^{\mathrm{th}}$ columns of $W$ and $V$.

However, this paper proposes a different weight normalization due to the fact that the above method does not normalize the mean value of the input:

$$y = ReLU(\frac{w^t x}{w} - \alpha) + \alpha$$

where $\alpha$ is a learnable parameter.

Note that the activation used here is called "Translated ReLU (TReLU)":

$$TReLU_\alpha(x) = \begin{cases} x & x \geq \alpha \\ \alpha & x < \alpha \end{cases}$$

# Evaluation Method

The evaluation method suggested here is inspired from the reconstruction error evaluation method of VAEs. The reconstruction loss of a Generator in GANs on a test set $X = \{x^1, ..., x^m\}$ is defined as:

$$\mathcal{L}_{rec}(G, X) = \frac{1}{m} \sum_{i=1}^{m} \min_z ||G(z) - x^i||_2^2$$

There is no way to infer $z$ from $x$ in a GAN setup so, after initializing $z = 0$, the following is used to get the best possible reconstruction:

1. Using current $z$ get $G(z)$
2. Backprop and update $z$ using L2 loss between $G(z)$ and $X$

   Repeat till convergence