Supplementary Material: Improving Gradient Regularization using Complex-Valued Neural Networks

Anonymous

February 2021

1 Gradient Constraint Derivation

$$\underline{x}_R = \cos(\underline{x}_i)$$

$$\underline{x}_I = \sin(\underline{x}_i)$$

$$g_i(\underline{x}_i) = \left[(W_R\underline{x}_R - W_I\underline{x}_I + \underline{b}_R)^2 + (W_R\underline{x}_I + W_I\underline{x}_R + \underline{b}_I)^2\right]^{\frac{1}{2}} = \sqrt{\underline{z}_R^2 + \underline{z}_I^2},$$

$$\frac{\partial g_i(\underline{x}_i)}{\partial W_R} = \frac{\underline{z}_R\underline{x}_R^T + \underline{z}_I\underline{x}_I^T}{g_i(\underline{x}_i)}$$
$$\frac{\partial g_i(\underline{x}_i)}{\partial W_I} = \frac{\underline{z}_I\underline{x}_R^T - \underline{z}_R\underline{x}_I^T}{g_i(\underline{x}_i)}$$

$$\begin{aligned}
(\frac{\partial g_i(\underline{x}_i)}{\partial W_R})^2 + (\frac{\partial g_i(\underline{x}_i)}{\partial W_I})^2 &= \left(\frac{\underline{z}_R\underline{x}_R^T + \underline{z}_I\underline{x}_I^T}{g_i(\underline{x}_i)}\right)^2 + \left(\frac{\underline{z}_I\underline{x}_R^T - \underline{z}_R\underline{x}_I^T}{g_i(\underline{x}_i)}\right)^2 \\
&= \frac{\underline{z}_R^2\underline{x}_R^{T^2} + 2\underline{z}_R\underline{z}_I\underline{x}_R^T\underline{x}_I^T + \underline{z}_I^2\underline{x}_I^{T^2} + \underline{z}_I^2\underline{x}_R^{T^2} - 2\underline{z}_R\underline{z}_I\underline{x}_R^T\underline{x}_I^T + \underline{z}_R^2\underline{x}_I^{T^2}}{g_i(\underline{x}_i)^2} \\
&= \frac{\underline{z}_R^2\underline{x}_R^{T^2} + \underline{z}_I^2\underline{x}_I^{T^2} + \underline{z}_I^2\underline{x}_R^{T^2} + \underline{z}_R^2\underline{x}_I^{T^2}}{g_i(\underline{x}_i)^2} \\
&= \frac{(\underline{z}_R^2 + \underline{z}_I^2)(\underline{x}_I^{T^2} + \underline{x}_R^{T^2})}{g_i(\underline{x}_i)^2} = \frac{g_i(\underline{x}_i)^2(\cos(\underline{x}_i)^2 + \sin(\underline{x}_i)^2)^T}{g_i(\underline{x}_i)^2} \\
&= \frac{g_i(\underline{x}_i)^2}{g_i(\underline{x}_i)^2} = 1
\end{aligned}$$

2 Additional Data

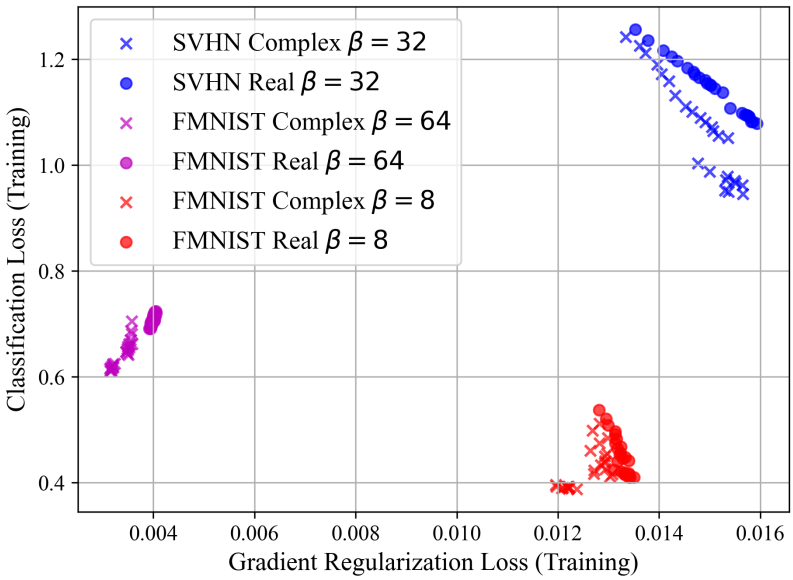

Figure 1: Classification Loss vs Gradient Regularized Loss for the FashionMNIST and SVHN datasets.

Comparison of training times. The training times of the networks used in the paper for MNIST are 253s, 336s, 585s, and 921s for real-valued nets trained with no defense, grad. reg., 4-step adv. train., and 8-step adv. train., respectively. It takes the complex-valued nets 360s and 604s for training with no defense and grad. reg., respectively. All experiments are run with PyTorch (Paszke et al 2017) on an Nvidia RTX 2070 Super GPU. We attribute the higher CVNN training time to our relatively unoptimized complex-valued PyTorch implementation. Finlay and Oberman (2021) present an approximation of double-backpropagation that could lead to a faster gradient regularization implementation than was used for our experiments.

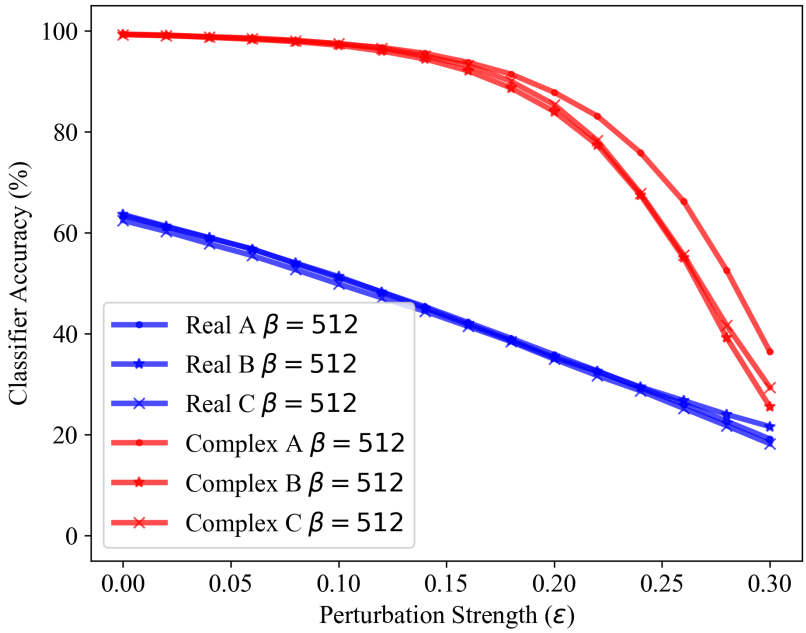

Figure 2: White box PGD(8) attacks against multiple independently trained networks on the MNIST dataset. Attacks are performed using the test set. Multiple networks (A, B, C) are trained with $\beta = 512$ gradient regularization on the MNIST dataset.

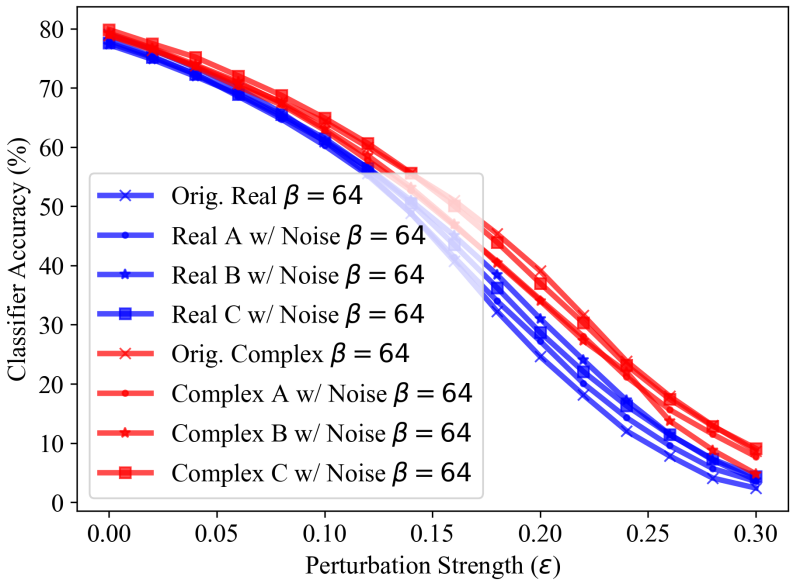

Figure 3: White box PGD(8) attacks against multiple independently trained networks on the FashionMNIST dataset. Attacks are performed using the test set. Multiple networks (A, B, C) are trained with $\beta = 64$ gradient regularization and $\mathcal{N}(\mu = 0, \sigma = 0.05)$ additive Gaussian noise. "Orig." denotes the network that was originally used in the paper.