Parseval Networks: Improving Robustness to Adversarial Examples

Moustapha Cisse[1], Piotr Bojanowski[1], Edouard Grave[1], Yann Dauphin[1], Nicolas Usunier[1]

[1]Facebook AI Research

ICML, 2017/ Presenter: Anant Kharkar

Outline

1. Introduction
   - Motivation
2. Theoretical Background
   - Formalization
   - Lipschitz Constant
   - Generalization Error
3. Parseval Networks
   - Basics
   - Results

Outline

1. Introduction
   - Motivation
2. Theoretical Background
   - Formalization
   - Lipschitz Constant
   - Generalization Error
3. Parseval Networks
   - Basics
   - Results

Motivation

- Neural networks achieve extreme accuracy on image classification tasks...
- ...but are vulnerable to adversarial images
- Regularization is ineffective
- Current approaches: distillation (Papernot et al., 2016) adversarial training (Goodfellow et al., 2015)

Contribution: regularization-based approach to adversarial robustness
Objective: minimize Lipschitz constant

Outline

1. Introduction
   - Motivation
2. Theoretical Background
   - Formalization
   - Lipschitz Constant
   - Generalization Error
3. Parseval Networks
   - Basics
   - Results

Basic Formalization

Neural network node:

$$n : x \mapsto \phi^{(n)}(W^{(n)}, (n'(x))_{n':(n,n') \in \varepsilon})$$

ϕ: activation, n': previous node
Final neural net output: $g(x, W)$

Adversarial example:

$$\tilde{x} = \underset{\tilde{x}:\|\tilde{x}-x\|_p \leq \epsilon}{argmax} \, (\ell(g(\tilde{x}, W), y)$$

Outline

1 Introduction
- Motivation

2 Theoretical Background
- Formalization
- Lipschitz Constant
- Generalization Error

3 Parseval Networks
- Basics
- Results

Lipschitz Constant

Objective: minimize the Lipschitz constant

Assume that:

$$\forall z, z' \in \mathbb{R}^Y, \forall \bar{y} \in \mathcal{Y}:$$

$$|\ell(z, \bar{y}) - \ell(z', \bar{y})| \leq \lambda_p \|z - z'\|_p$$

Or alternatively:

$$\frac{|\ell(z, \bar{y}) - \ell(z', \bar{y})|}{\|z - z'\|_p} \leq \lambda_p$$

Thus, Lipschitz constant λ_p is a bound on the magnitude of the point-wise slope of the loss

Outline

1. Introduction
   - Motivation
2. Theoretical Background
   - Formalization
   - Lipschitz Constant
   - Generalization Error
3. Parseval Networks
   - Basics
   - Results

Generalization Error

Basic error:

$$L(W) = \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}[\ell(g(x, W), y)]$$

Adversarial error:

$$L_{adv}(W, p, \epsilon) = \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}[\max_{\tilde{x}:\|\tilde{x}-x\|\leq\epsilon} \ell(g(x, W), y)]$$

We know that $L(W) \leq L_{adv}(W, p, \epsilon)$

Generalization Error

Basic error:

$$L(W) = \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}[\ell(g(x, W), y)]$$

Adversarial error:

$$L_{adv}(W, p, \epsilon) = \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}[\max_{\tilde{x}:\|\tilde{x}-x\|\leq\epsilon} \ell(g(x, W), y)]$$

We know that $L(W) \leq L_{adv}(W, p, \epsilon)$

$$\begin{aligned} L_{adv}(W, p, \epsilon) &\leq L(W) + \\ &\quad \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}[\max_{\tilde{x}:\|\tilde{x}-x\|\leq\epsilon} |\ell(g(\tilde{x}, W), y) - \ell(g(x, W), y)|] \\ &\leq L(W) + \lambda_p \Lambda_p \epsilon \end{aligned}$$

Thus, $\lambda_p \Lambda_p \epsilon$ bounds added adversarial error

Lipschitz Constant of Neural Network

Perturbation based on previous layer:

$$\|n(x) - n(\tilde{x})\|_p \leq \sum_{n':(n,n')\in\varepsilon} \Lambda_p^{(n,n')} \|n'(x) - n'(\tilde{x})\|_p$$

Lipschitz constant in terms of previous layer:

$$\Lambda_p^{(n)} \leq \sum_{n':(n,n')\in\varepsilon} \Lambda_p^{(n,n')} \Lambda_p^{(n')}$$

Linear layers (2-norm):

$$\Lambda_2^{(n)} = \|W^{(n)}\|_2 \Lambda_2^{(n')}$$

$\|W^{(n)}\|_2$: spectral norm

Outline

1. Introduction
   - Motivation
2. Theoretical Background
   - Formalization
   - Lipschitz Constant
   - Generalization Error
3. Parseval Networks
   - Basics
   - Results

Basic Concept

Regularization to constrain Lipschitz constant of each hidden layer
Two concepts:

- Orthonormal rows in linear/convolutional layers
  - Required to control spectral norm
  - Minimize spectral norm to minimize Lipschitz constant
- Convex combinations in aggregation layers

Approximation

Optimize weights while maintainin orthogonality - requires approximation of orthogonality Enforce W as Parseval tight frame
Regularizer:

$$R_\beta(W_k) = \frac{\beta}{2} \| W_k^T W_k - \mathcal{I} \|_2^2$$

Weight update (2^{nd} step):

$$W_k \leftarrow (1+\beta) W_k - \beta W_k W_k^T W_k$$

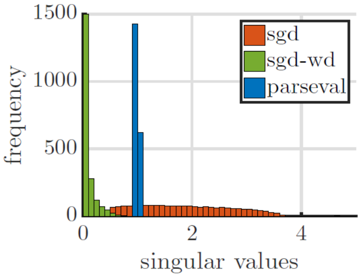
Outline

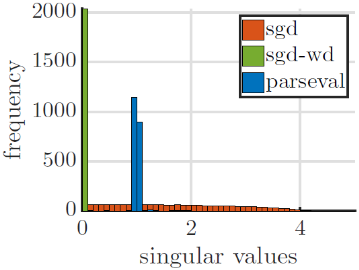
1 Introduction
- Motivation

2 Theoretical Background
- Formalization
- Lipschitz Constant
- Generalization Error

3 Parseval Networks
- Basics
- Results

Checking Orthogonality



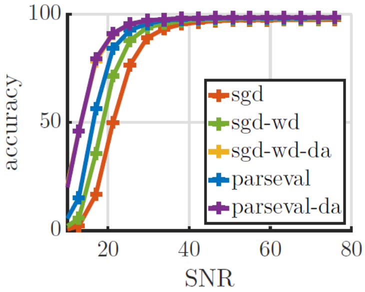


Layer 1

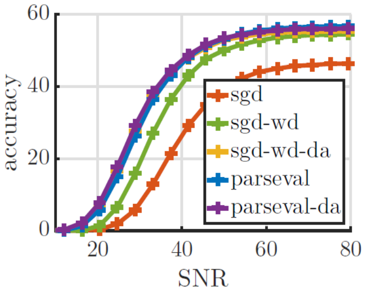
Layer 4

Singular values concentrated around 1 $\rightarrow$ quasi-orthogonal

Accuracy - Fully Connected Nets



MNIST



CIFAR-10

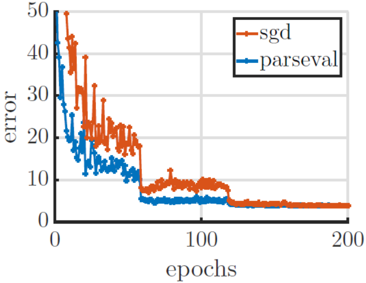
Parseval networks perform better at all SNRs

Accuracy - Residual Nets

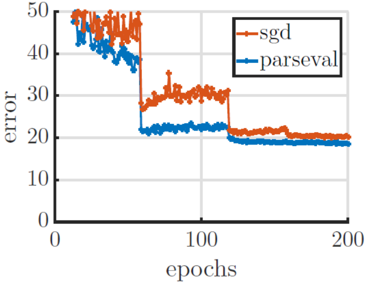
	Model	Clean	$\epsilon \approx 50$	$\epsilon \approx 45$	$\epsilon \approx 40$	$\epsilon \approx 33$
CIFAR-10	Vanilla	95.63	90.16	85.97	76.62	67.21
	Parseval(OC)	95.82	91.85	88.56	78.79	61.38
	Parseval	**96.28**	**93.03**	**90.40**	**81.76**	**69.10**
	Vanilla	95.49	91.17	88.90	86.75	84.87
	Parseval(OC)	95.59	92.31	90.00	**87.02**	**85.23**
	Parseval	**96.08**	**92.51**	**90.05**	86.89	84.53
CIFAR-100	Vanilla	79.70	65.76	57.27	44.62	34.49
	Parseval(OC)	81.07	70.33	63.78	49.97	32.99
	Parseval	**80.72**	**72.43**	**66.41**	**55.41**	**41.19**
	Vanilla	79.23	67.06	62.53	56.71	51.78
	Parseval(OC)	**80.34**	69.27	62.93	53.21	**52.60**
	Parseval	80.19	**73.41**	**67.16**	**58.86**	39.56
SVHN	Vanilla	**98.38**	97.04	95.18	92.71	88.11
	Parseval(OC)	97.91	**97.55**	**96.35**	**93.73**	**89.09**
	Parseval	98.13	97.86	96.19	93.55	88.47

Dimensionality & Convergence

	SGD-wd		SGD-wd-da		Parseval	
	all	class	all	class	all	class
Layer 1	72.6	34.7	73.6	34.7	89.0	38.4
Layer 2	1.5	1.3	1.5	1.3	82.6	38.2
Layer 3	0.5	0.5	0.4	0.4	81.9	30.6
Layer 4	0.5	0.4	0.4	0.4	56.0	19.3



CIFAR-10



CIFAR-100