Predicting Under and Overfitting in Deep Neural Networks (DNNs) using Graph Smoothness

Carlos Lassance, Vincent Gripon, Antonio Ortega

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

USC University of Southern California

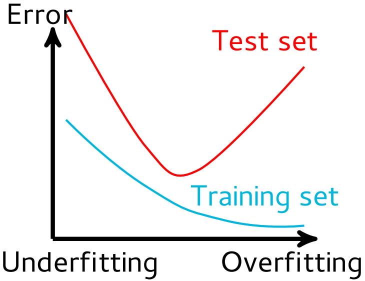
June 7, 2018

Graph Signal Processing Workshop 2018

Introduction

DNNs can approximate **any** function:

- +: Perfect fit is achievable;
- -: **Training** performance $\not\Rightarrow$ **Test** performance.



Standard solution: Cross Validation.

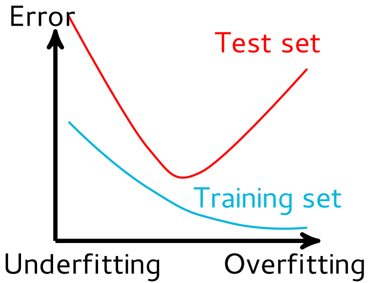
Objective

- Better understand what determines generalization in DNNs.

Introduction

DNNs can approximate **any** function:

- +: Perfect fit is achievable;
- -: **Training** performance $\not\Rightarrow$ **Test** performance.



Standard solution: Cross Validation.

Objective

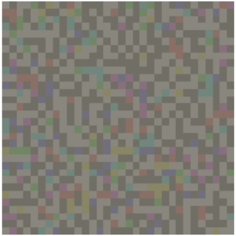
- Better understand what determines generalization in DNNs.

Adversarial Examples

Definition

A noisy image that:

- Has high signal-to-noise ratio (SNR);
- **Fools** a classifier.

Original	Noise	Adversarial Image
		
Deer 99.96%	Cat 36.63%	Cat 90.66%

Definitions

Graph Construction - Adjancency Matrix (A)

- Batch with M examples from each class,
- Generate intermediate representations for layer ℓ,
- Create a pairwise cosine similarity matrix,
- Threshold the k nearest neighbors.

Laplacian

$$L = D - A$$

Label signal and Smoothness

We consider here label signals $\mathbf{s}$: indicator vectors of classes.

$$\mathbf{s}^\top L \mathbf{s} = \sum_{i=1}^{d} \Lambda_{ii} \hat{\mathbf{s}}_i^2 = \sum_{u \leftrightarrow v} A_{uv} (\mathbf{s}_u - \mathbf{s}_v)^2 .$$

Definitions

Graph Construction - Adjancency Matrix (A)

- Batch with M examples from each class,
- Generate intermediate representations for layer ℓ,
- Create a pairwise cosine similarity matrix,
- Threshold the k nearest neighbors.

Laplacian

$$L = D - A$$

Label signal and Smoothness

We consider here label signals $\mathbf{s}$: indicator vectors of classes.

$$\mathbf{s}^\top L \mathbf{s} = \sum_{i=1}^{d} \Lambda_{ii} \hat{\mathbf{s}}_i^2 = \sum_{u \leftrightarrow v} A_{uv} (\mathbf{s}_u - \mathbf{s}_v)^2 .$$

Definitions

Graph Construction - Adjancency Matrix (A)

- Batch with M examples from each class,
- Generate intermediate representations for layer ℓ,
- Create a pairwise cosine similarity matrix,
- Threshold the k nearest neighbors.

Laplacian

$$L = D - A$$

Label signal and Smoothness

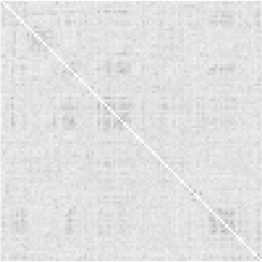
We consider here label signals $\mathbf{s}$: indicator vectors of classes.

$$\mathbf{s}^\top L \mathbf{s} = \sum_{i=1}^{d} \Lambda_{ii} \hat{\mathbf{s}}_i^2 = \sum_{u \leftrightarrow v} A_{uv} (\mathbf{s}_u - \mathbf{s}_v)^2 \ .$$

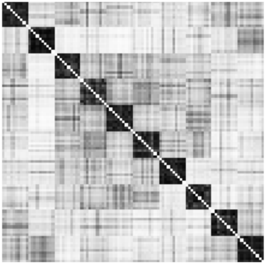
Laplacian example

To illustrate what our Laplacian represents:

- Trained network + batch of examples ordered by class,
- Generate a complete graph,
- Dark points are highly correlated.



Middle layer



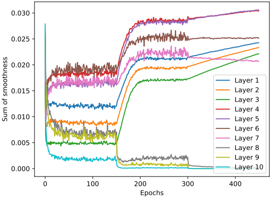
Deep layer

Prior Work

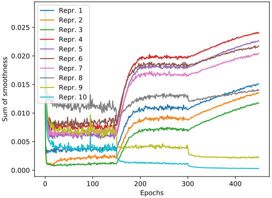
An Inside Look at DNNs using Graph Signal Processing

- Gripon, Ortega, Girault, 2018 at ITA,
- Analyses how the smoothness evolves over the network,
- Difference in behavior for under/over/ok networks.

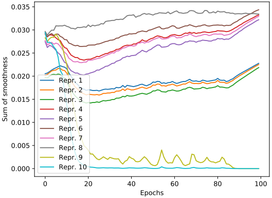
Baseline



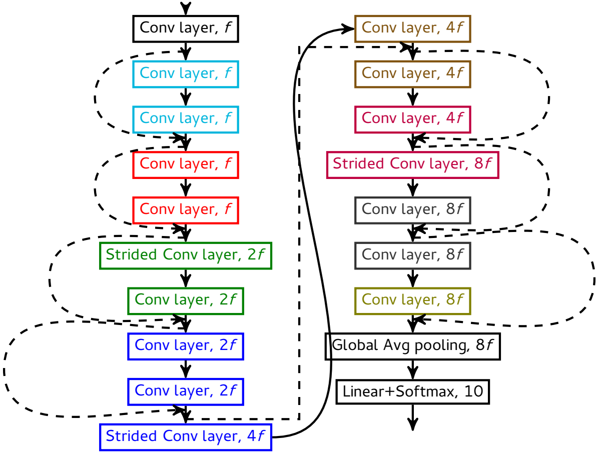
Underfitting



Random Labels



Studied architecture



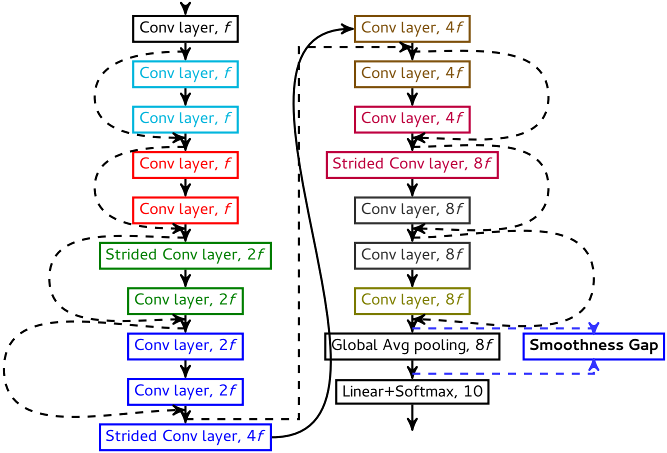
Experiments

Setup

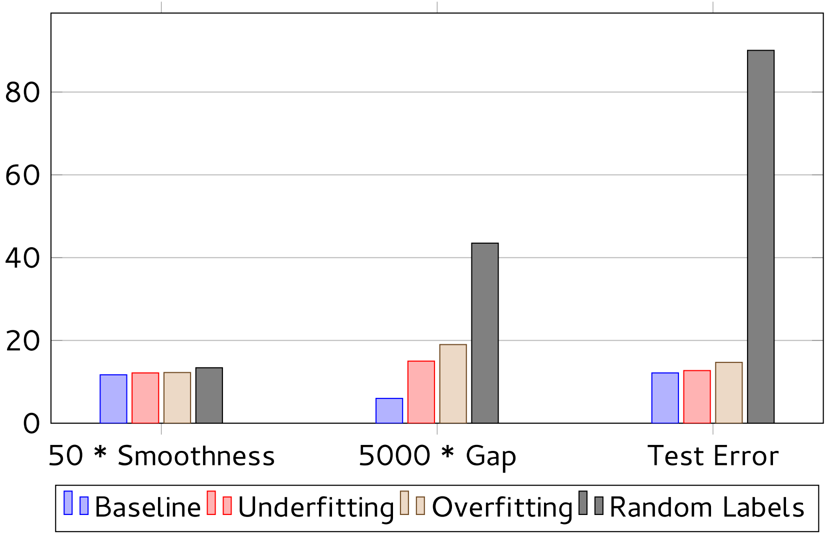
- We test the same architecture with different hyperparameters.
- To stress generalization to:
  - **Underfitting**: Different scales of network size;
  - **Overfitting**: Different sizes of the training set;
  - **Architecture scale**: Different types of network scale.

- Smoothness gap between:
  - before the global average pooling;
  - after the global average pooling.
- Graphs of 500 examples from the **training set**,
- Thresholded by the 20 nearest neighbors,
- We evaluate the measure by the mean gap over 10 graphs.

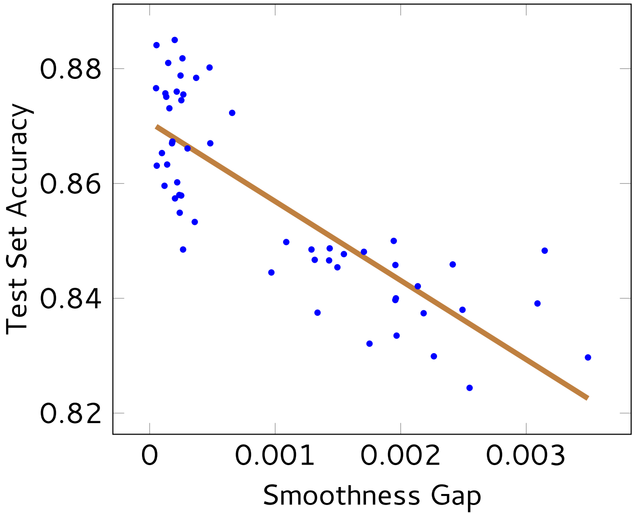
Studied architecture



Results, Different states



Results, Network Scale



Results, R^2

Reported value

R^2 value of a **linear regression** of given measure against the test set.

Results

Case/Measure	Training Accuracy	Network Size	**Ours**
Underfitting	54%	50%	**84%**
Overfitting	19%	Not Applicable	**68%**
Network Scale	30%	14%	**67%**

Can we use smoothness gap as a regularizer during training?

Results, R^2

Reported value

R^2 value of a **linear regression** of given measure against the test set.

Results

Case/Measure	Training Accuracy	Network Size	**Ours**
Underfitting	54%	50%	**84%**
Overfitting	19%	Not Applicable	**68%**
Network Scale	30%	14%	**67%**

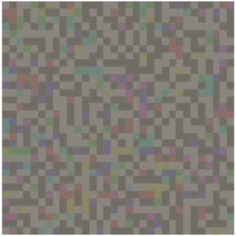
Can we use smoothness gap as a regularizer during training?

Adversarial Examples

Definition

A noisy image that:

- Has high signal-to-noise ratio (SNR);
- **Fools** a classifier.

Original	Noise	Adversarial Image
		
Deer 99.96%	Cat 36.63%	Cat 90.66%

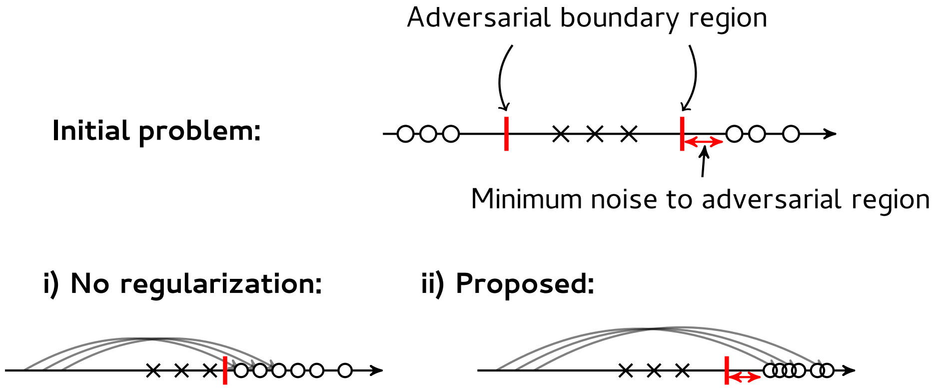
Recent work in defense against adversarial noise

Parseval Networks, Cisse et al, ICML 2017

- Bounding the Lipschitz constant of layers.
- Distance between 2 examples can only decrease in the network:
  - Interested in the effects of the weights,
  - As the distance is small at the start it should be small at the end,
  - Similar to our smoothness metric,
  - Examples domain vs Class domain,

Our Proposal

- Bound the smoothness gap between successive layers.

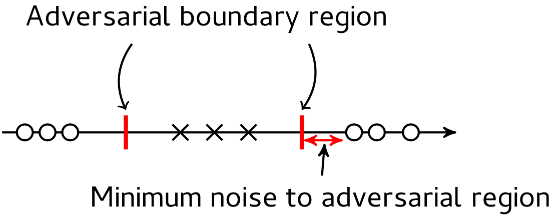


- Laplacian Power Networks:
  Bounding Indicator Function Smoothness for Adversarial Defense
  - https://arxiv.org/pdf/1805.10133.pdf

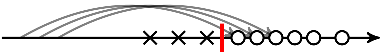
Our Proposal

- Bound the smoothness gap between successive layers.

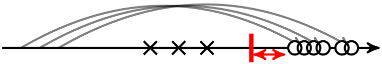
Initial problem:



i) No regularization:



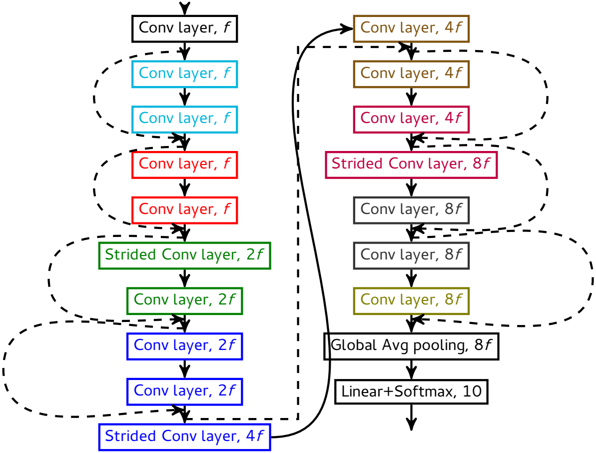
ii) Proposed:



- Laplacian Power Networks:
  Bounding Indicator Function Smoothness for Adversarial Defense
  - https://arxiv.org/pdf/1805.10133.pdf

Studied architecture

Bound the sum of absolute smoothness gap between all layers

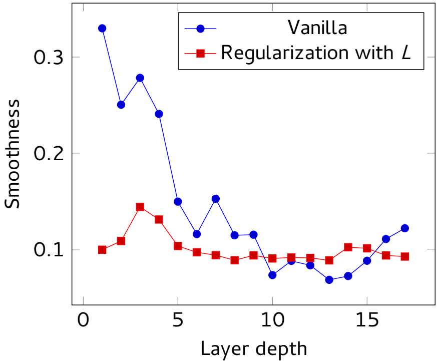


Results

Defense/SNR	Clean	50	33
None	88.47%	80.10%	33.25%
Parseval	**89.87%**	**83.06%**	45.11%
Ours	87.25%	82.35%	**50.16%**
Both	89.08%	82.52%	**50.25%**

Results

Defense/SNR	Clean	50	33
None	88.47%	80.10%	33.25%
Parseval	**89.87%**	**83.06%**	45.11%
Ours	87.25%	82.35%	**50.16%**
Both	89.08%	82.52%	**50.25%**



Conclusion

- Smoothness of label signals seem interesting to control generalization in DNNs:
  - Can be used to identify interesting architectures,
  - Can be used to limit adversarial noise

Future Work

- Need for theoretical understanding,
- Better construction of graphs?
- Use of graph sampling techniques?

Conclusion

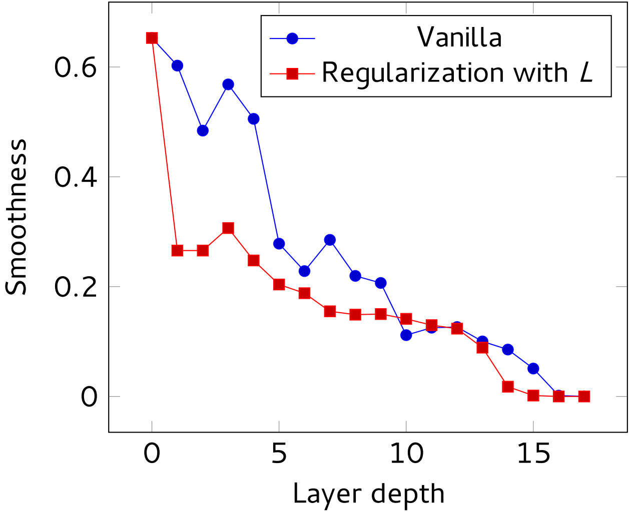
Conclusion

- Smoothness of label signals seem interesting to control generalization in DNNs:
  - Can be used to identify interesting architectures,
  - Can be used to limit adversarial noise

Future Work

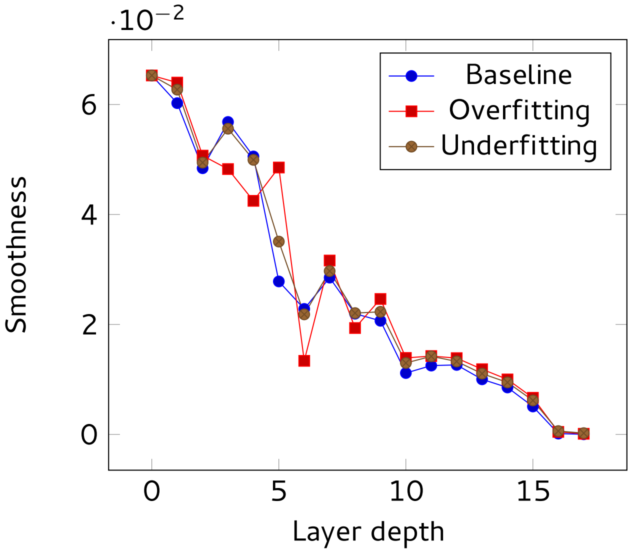
- Need for theoretical understanding,
- Better construction of graphs?
- Use of graph sampling techniques?

Results



Vanilla
Regularization with L
Smoothness
0.6
0.4
0.2
0
0
5
10
15
Layer depth

Results



$\cdot 10^{-2}$
Baseline
Overfitting
Underfitting
Smoothness
0
2
4
6
0
5
10
15
Layer depth