Introducing Graph Smoothness Loss for Training Deep Learning Architectures

Myriam Bontonou, **Carlos Lassance**, Ghouthi Boukli Hacene, Vincent Gripon, Jian Tang, Antonio Ortega

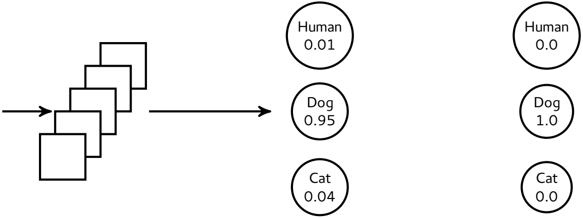
Mila (UdeM and HEC), IMT Atlantique and USC

SDM'19 Workshop on Deep Learning for Graphs

May 4th, 2019

Supervised classification with DNNs

State of the art on various tasks, from NLP to computer vision:



Input: x

Parameters: θ

$\hat{y} = f(x, \theta)$

Targets: y

Goal: Search θ to minimize Cross Entropy (CE) between y and $\hat{y}$

Motivation

Problems

- Forces all examples of the same class to have the same output:
  - Highly-deformed space may lead to less robust classification;
- Choses arbitrarly the targets (one-hot embedding);
  - Disregards initialization and inputs;
- Output dimension is equal to the number of classes:
  - Problem in continual learning;

Desired Properties

- **Property No Collapse:**
  - Do not collapse the outputs of the same class.
- **Property Learned Outputs:**
  - Output must be chosen by the learning algorithm.
- **Property Arbitrary Output Size:**
  - Output dimension must be an hyperparameter.

Motivation

Problems

- Forces all examples of the same class to have the same output:
  - Highly-deformed space may lead to less robust classification;
- Choses arbitrarly the targets (one-hot embedding);
  - Disregards initialization and inputs;
- Output dimension is equal to the number of classes:
  - Problem in continual learning;

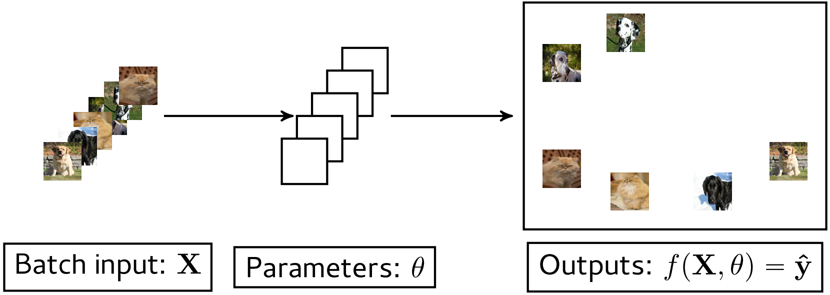
Desired Properties

- **Property No Collapse:**
  - Do not collapse the outputs of the same class.
- **Property Learned Outputs:**
  - Output must be chosen by the learning algorithm.
- **Property Arbitrary Output Size:**
  - Output dimension must be an hyperparameter.

Works		Properties		
Method	Reference	No Collapse	Learned Outputs	Arbitrary Output Size
One-hot embedding	—			
Distillation	[Hinton et al 2015]		X	
Error correcting codes	[Dietterich & Bakiri 1994]			X
Triplet Loss	[Hoffer & Aillon 2015]		X	X
Smoothness		**X**	**X**	**X**

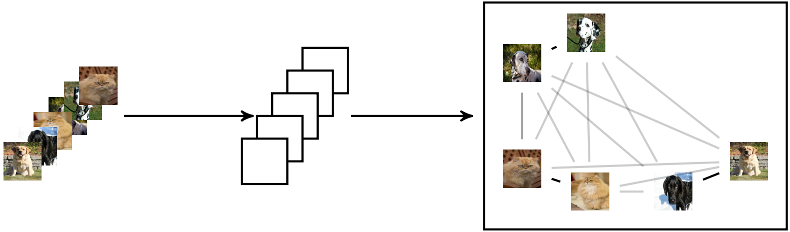
Smoothness cost function

Setup



Smoothness cost function

Graph inference



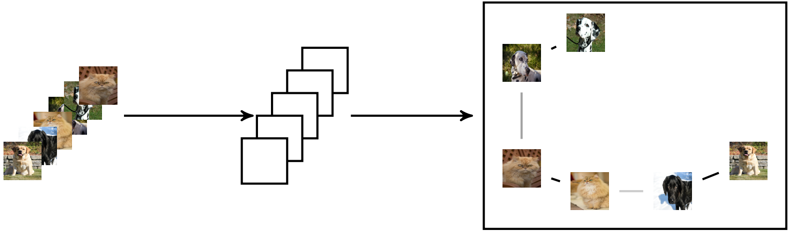
Batch input: $\mathbf{X}$

Parameters: θ

Graph: $G = (V, W), V = \hat{\mathbf{y}}$

Goal: Search θ so that the class labels are smooth on the graph G

Smoothness cost function
Graph inference



Batch input: $\mathbf{X}$

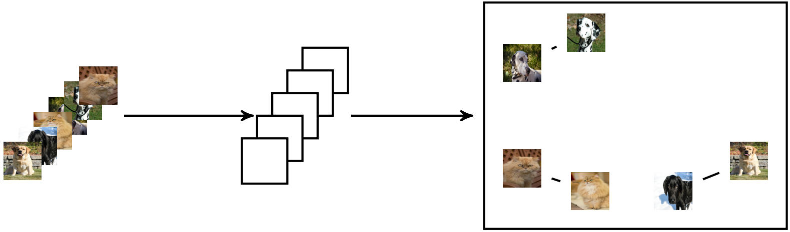
Parameters: θ

Graph: $G = (V, W), V = \hat{\mathbf{y}}$

Goal: Search θ to minimize graph smoothness of a signal s over the graph G

Smoothness cost function

Graph inference



Batch input: $\mathbf{X}$

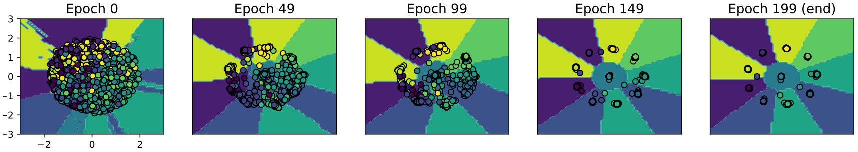
Parameters: θ

Graph: $G = (V, W), V = \hat{\mathbf{y}}$

Goal: Search θ to minimize graph smoothness of a signal s over the graph G

Experiments

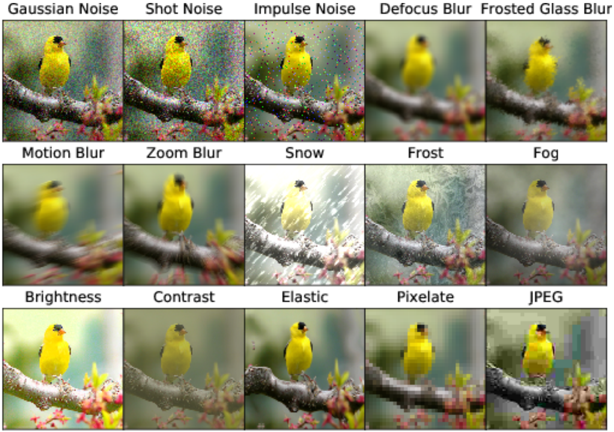
2D Visualization CIFAR-10 dataset



- Sanity check comparison between CE and Smoothness;

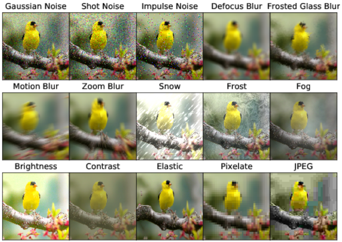
Loss	Classifier	CIFAR-10	CIFAR-100	SVHN
Cross-entropy	Argmax	**5.06%**	**27.92%**	3.69%
Smoothness	1-NN	5.63%	29.17%	3.84%
Smoothness	10-NN	5.48%	28.82%	**3.34**%
Smoothness	RBF SVC	5.50%	30.55%	3.40%

- Robustness benchmark [Heynckes & Dietterich 2019];
- Relative performance to a baseline.



Experiments

Robustness



Cost function	Clean test error	MCE	relative MCE
Cross-entropy	**5.06%**	100	100
Smoothness	5.63%	**95.28**	**90.33**

Conclusion and Future work

Conclusion

- Similar performance to cross entropy;
- More degrees of freedom;
- Increased Robustness.

Future work

- Increase performance on clean settings;
- Stronger link between loss function and classification algorithm;
- Continual training.

Extended article available at: `http://arxiv.org/abs/1905.00301`.

Smoothness cost function

Graph signal processing background

Graph

Based on the similarities between the outputs of the network.

- $G = \langle V, \mathbf{W} \rangle$
- $\mathbf{W}(\mu, \nu) = \exp\left(-\alpha \| f(\mathbf{x}[\mu]) - f(\mathbf{x}[\nu]) \|\right)$
- $x[\mu], x[\nu] \in V$

Graph signal smoothness

$$\sigma(G, \mathbf{s}) = \mathbf{s}^\top \mathbf{L} \mathbf{s} = \sum_{x[\mu], x[\nu] \in V} \sum \mathbf{W}[\mu, \nu] \left(\mathbf{s}[\mu] - \mathbf{s}[\nu]\right)^2 , \quad (1)$$

- $\mathbf{L}$: Laplacian operator ($\mathbf{L} = \mathbf{D} - \mathbf{W}$);
- s: Signal on the graph.
  - In this work equivalent to an one-hot embedding (y).

Smoothness cost function

Cost function

Sum of similarities between elements of different classes:

$$\mathcal{L}_{smoothness}(f, V) = \sum_{\substack{\mathbf{x}[\mu],\mathbf{x}[\nu] \in V \\ \mathbf{y}[\mu]\mathbf{y}[\nu]=0}} \exp\left(-\alpha \| f(\mathbf{x}[\mu]) - f(\mathbf{x}[\nu]) \|\right) .$$

Cross entropy + One hot embedding

Supervised Classification

- Loss: Categorical cross entropy;
- Output: One-hot embedding.

Categorical cross entropy

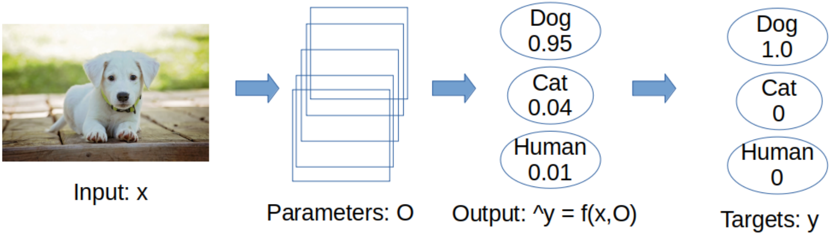
$$\mathcal{L}_{ce}(f, \mathcal{D}) = \sum_{(x,y)\in\mathcal{D}} \sum_{i=0}^{c} y_i log\left(f(x)_i\right)$$

One-hot embedding

$$y_i = \begin{cases} 1, & \text{if } i = c \\ 0, & \text{otherwise} \end{cases}$$

Supervised classification with DNNs

State of the art on various tasks, from NLP to computer vision:



Goal: Optimize O to minimize ylog(^y)